THE SURROGATE MATRIX METHODOLOGY: A PRIORI ERROR ESTIMATION

DANIEL DRZISGA*, BRENDAN KEITH*, AND BARBARA WOHLMUTH*

Abstract. We give the first mathematically rigorous analysis of an emerging approach to finite element analysis (see, e.g., Bauer et al. [Appl. Numer. Math., 2017]), which we hereby refer to as *the surrogate matrix methodology*. This methodology is based on the piece-wise smooth approximation of the matrices involved in a standard finite element discretization. In particular, it relies on the projection of smooth so-called stencil functions onto high-order polynomial subspaces. The performance advantage of the surrogate matrix methodology is seen in constructions where each stencil function uniquely determines the values of a significant collection of matrix entries. Such constructions are shown to be widely achievable through the use of locally-structured meshes. Therefore, this methodology can be applied to a wide variety of physically meaningful problems, including nonlinear problems and problems with curvilinear geometries. Rigorous *a priori* error analysis certifies the convergence of a novel surrogate method for the variable coefficient Poisson equation. The flexibility of the methodology is also demonstrated through the construction of novel methods for linear elasticity and nonlinear diffusion problems. In numerous numerical experiments, we demonstrate the efficacy of these new methods in a matrix-free environment with geometric multigrid solvers. In our experiments, up to a twenty-fold decrease in computation time is witnessed over the classical method with an otherwise identical implementation.

Key words. Surrogate numerical methods, finite element methods, matrix-free, high performance computing, a priori analysis, low order, geometric multigrid.

AMS subject classifications. 65D05, 65M60, 65N30, 65Y05, 65Y20.

1. Introduction.

In the field of computational science, major funding initiatives in North America, Europe, and Asia have thrust high performance computing (HPC) to ascendancy. In anticipation of future exascale computers, much work in this discipline involves the deep and careful reconstruction of long-established computing practices. An important characteristic of numerical algorithms aimed for these computers, is the floating-point operation (FLOP) per byte ratio. In order to achieve optimal performance and power efficiency on future machines, the time spent on FLOPs relative to memory transfer needs to be substantial.

Most traditional finite element softwares assemble global stiffness matrices by looping over elements and adding the corresponding local contributions to the global matrix. Storing the resulting sparse matrices requires significantly more memory than just storing the degrees of freedom. However, memory consumption is certainly not the only obstacle at the computational frontier. Indeed, at such scales, the memory traffic and latency involved in loading indices and entries for matrix vector products (MVPs) also presents critical challenges.

Since iterative solvers only require MVPs, it is not necessary to store all of the nonzeros of the global matrix in memory. Instead, it is sufficient to compute the nonzero entries on-the-fly, i.e., matrix-free. Different tactics exist to implement matrix-free methods, but the predominant candidate for low-order finite elements is the element-by-element approach [3, 15, 19, 24, 38], wherein local stiffness matrices are multiplied by local vectors and later added to the global solution vector. These local stiffness matrices may either be stored in memory—which actually requires more memory than storing the global matrix—or computed on-the-fly. When using high-order finite elements, the weak forms can be integrated on-the-fly using standard or reduced quadrature formulas [17, 30, 31, 32, 34]. This is a well-suited tactic for future machines because of its large arithmetic intensity [33].

Significant performance gains are often attributed to a problem, scale, and architecture-specific balance between FLOPs and memory traffic. As a matter of course, exploiting symmetries in a problem or discretization can significantly improve the time to solution. This is often the

*Lehrstuhl für Numerische Mathematik, Fakultät für Mathematik (M2), Technische Universität München, Garching bei München (drzisga@ma.tum.de, keith@ma.tum.de, wohlmuth@ma.tum.de)

cause of enormous speed-ups in computations on structured meshes. Likewise, high performance of matrix-free methods can be most easily achieved in homogeneous problems with simple geometries. Nevertheless, most geometries coming from significant real-world problems cannot be adequately approximated by fully-structured meshes. A possible trade-off is to use locally-structured meshes like hierarchical hybrid grids (HHG) where initially unstructured coarse grids are locally refined in a uniform way. This local structure allows the application of stencil-based finite element procedures which operate similar to finite difference methods. By using these grids, efficient stencil-based methods have been successfully applied to a wide range of problems [11, 12, 13, 22, 26]. A related approach, suitable for low-order finite element discretizations of scalar elliptic partial differential equations (PDEs) with variable coefficients, based on scaling of reference stencils is discussed in [7].

In this paper, we revisit the classical lowest-order Bubnov–Galerkin finite element method and analyze a modification of it which is strongly amenable to stencil-based matrix-free computation. In our approach, a macro-mesh, which is not required to have any global structure, is used to triangulate the model geometry. This macro-mesh is then uniformly refined a large number of times, resulting in a fine-scale locally-structured mesh. For each macro-element, a *local approximation* of the fine-scale global matrix delivers a fine-scale *surrogate matrix* which maintains the convergence properties of the fine-scale discrete solution, up to the original order of the approximation. A related investigation [10] illustrated the promise of this methodology and provided numerical evidence for the convergence rates which are proven here rigorously. This work considered only Poisson's equation. Later on, Stokes flow (with variable viscosity) was considered in two follow-up articles [8, 9]. Each of these initial studies focused on the massively parallel high performance computing aspects of their respective methods. These studies used the HHG software framework [11, 12, 13] in their experiments. Here, the finite-element software framework HyTeG [29] is used.

In this paper, we recast the central features of the original work as a methodology complete with a mathematical framework suitable for rigorous analysis. The principal novelty is the mathematical foundation developed here, which can be used to analyze further incarnations of the methodology. In total, we consider three specific mathematical model problems: the variable coefficient Poisson equation, linear elastostatics, and p-Laplacian diffusion. Although our presentation demonstrates that the surrogate matrix methodology applies to each of these models equally well, we only employ a complete *a priori* analysis of the simplest model, the variable coefficient Poisson equation. In our numerical experiments, we carefully verify the proven *a priori* convergence rates with the variable coefficient Poisson equation. We also include proof-of-concept demonstrations from numerical experiments with the linear elastostatics and p-Laplacian diffusion problems.

2. Notation and outline. Let V be a reflexive Banach space over $\mathbb{R}$, the field of real numbers, and let $V_h \subsetneq V$ be a finite-dimensional subspace. Consider a continuous and weakly coercive bilinear form $a : V \times V \to \mathbb{R}$ and a bounded linear functional $F \in V^*$, the topological dual of V.

In this paper, we are concerned with the solutions u, u_h, and $\widetilde{u}_h$ of the following three abstract variational problems.

$$\text{(2.1a)} \qquad \text{Find } u \in V \text{ satisfying} \qquad a(u,v) = F(v) \qquad \text{for all } v \in V\,.$$

$$\text{(2.1b)} \qquad \text{Find } u_h \in V_h \text{ satisfying} \qquad a(u_h,v_h) = F(v_h) \qquad \text{for all } v_h \in V_h\,.$$

$$\text{(2.1c)} \qquad \text{Find } \widetilde{u}_h \in V_h \text{ satisfying} \qquad \widetilde{a}(\widetilde{u}_h,v_h) = F(v_h) \qquad \text{for all } v_h \in V_h\,.$$

In (2.1c), a *surrogate* bilinear form $\widetilde{a} : V_h \times V_h \to \mathbb{R}$ has been introduced. In order to properly define $\widetilde{a}(\cdot,\cdot)$, some additional assumptions on $a(\cdot,\cdot)$ are still required; see Section 3.

The discrete variational problems (2.1b) and (2.1c) induce matrix equations for coefficients

$\mathsf{u}, \widetilde{\mathsf{u}}$ in some $\mathbb{R}^N$,

$$\mathsf{A}\mathsf{u} = \mathsf{f} \qquad \text{and} \qquad \widetilde{\mathsf{A}}\widetilde{\mathsf{u}} = \mathsf{f}, \tag{2.2}$$

respectively. In the first case, fix a basis for V_h, say $\{\phi_i\}_{i=1}^N$. For this basis, each (i,j)-component of the stiffness matrix A is simply $\mathsf{A}_{ij} = a(\phi_j, \phi_i)$. In the following section, we present a methodology to construct a *surrogate* stiffness matrix $\widetilde{\mathsf{A}} \approx \mathsf{A}$ which can be used in place of the *true* stiffness matrix A. This methodology stands apart from technical details, such as differences in quadrature formulas. Section 4 provides a short (non-comprehensive) list of examples fitting into our framework. In Section 5, we discuss the incorporation of non-homogeneous boundary conditions and what we hereon refer to as *the zero row sum property*. In Section 6, sufficient conditions for the discrete stability of surrogate bilinear forms $\widetilde{a}(\cdot,\cdot)$ are briefly discussed. Next, in Section 7, we perform a rigorous *a priori* error analysis of our approach applied to the variable coefficient Poisson equation. A brief description of our implementation is given in Section 8. Then, in Section 9, we document several numerical experiments. Here, a thorough verification of each error estimate in Section 7 is given. This is complemented by performance measurements for the additional examples.

Throughout this article, we assume that $\Omega \subseteq \mathbb{R}^n$ is a bounded Lipschitz domain. For matrices $\mathsf{M} \in \mathbb{R}^{l\times m}$, define the ℓ^∞-, and max-norms, $\|\mathsf{M}\|_\infty = \max_i \sum_j |\mathsf{M}_{ij}|$ and $\|\mathsf{M}\|_{\max} = \max_{i,j} |\mathsf{M}_{ij}|$. Likewise, for any function $v : \Omega \to \mathbb{R}$, we will use the similar notation, $\|v\|_0$, $\|v\|_1$, and $\|v\|_2$, for the canonical $L^2(\Omega)$-, $H^1(\Omega)$-, and $H^2(\Omega)$-norms, respectively. When dealing with a subset $T \subseteq \Omega$, denote the related $L^2(T)$-, $H^1(T)$-, and $H^2(T)$-norms by $\|v\|_{0,T}$, $\|v\|_{1,T}$, and $|v|_{2,T}$, respectively. For any simplex T and integer $0 \leq q < \infty$, we denote the space of polynomials of degree at most q as $\mathcal{P}_q(T)$. All remaining notation will be defined as it arises.

3. Surrogate stiffness matrices. In this section, we present the constitutive elements of the surrogate matrix methodology. Our approach here is to gradually introduce the necessary concepts, all the while maintaining a clear sense of generality. In order to arrive at a tractable framework for our problems of interest, we gradually refine the presentation from general n-dimensional spaces to only $n = 1, 2$, or 3 and from general Banach spaces to only $W^{1,p}(\Omega)$ (or products thereof), where $1 < p < \infty$. The intention of proceeding in this way is to indicate that the methodology can be applied to an extremely broad set of problems and, specifically, to most problems where finite element methods are traditionally applied.

3.1. Preliminary assumptions. Given a bounded domain $\Omega \subseteq \mathbb{R}^n$, assume that the true bilinear form can be expressed as

$$a(u,v) = \int_\Omega G(x, u(x), v(x))\,\mathrm{d}x \qquad \text{for all } u, v \in V. \tag{3.1a}$$

Additionally, upon defining $\operatorname{supp}(u) = \{y \in \Omega : u(y) \neq 0\}$ for smooth functions, make the following sparsity assumption:

$$G(x, u(y), v(y)) = 0 \qquad \text{whenever } y \notin \operatorname{supp}(u) \cap \operatorname{supp}(v). \tag{3.1b}$$

These assumptions permit us to consider the discretization of most classical differential operators. Indeed, in the assumptions above, the integrand $G(x,u,v)$ may induce distributional derivatives on its second and third arguments. Meanwhile, the first argument can be identified with the spatial argument of any associated variable coefficients. For example, in the weak form of a Poisson-type equation, $-\operatorname{div}(K\nabla u) = f$, with a variable, symmetric positive-definite diffusion tensor $K(x)$ (cf. Subsection 4.1), we have the bilinear form

$$a_1(u,v) = \int_\Omega \nabla u(x)^\top K(x) \nabla v(x)\,\mathrm{d}x \qquad \text{for all } u, v \in V = H^1(\Omega)\,. \tag{3.2}$$

Here, taking any point $x \in \Omega$, the integrand in (3.1a) reduces to $G(x,u,v) = \nabla u^\top K(x) \nabla v$. Evidently, this G satisfies the sparsity assumption (3.1b).

3.2. Stencil functions. Let $\phi \in V$ be a test function with compact support in Ω and, for any fixed $y \in \mathbb{R}^n$, define $\phi_y(x) = \phi(x-y)$. Now, consider any fixed set of ordered points $\mathbb{X} = \{x_i\}$ in Ω and recall (3.1a). Assuming that both $\phi_{x_i}, \phi_{x_j} \in V$, observe (via a simple change of variables) that

$$a(\phi_{x_j}, \phi_{x_i}) = \int_\Omega G(y, \phi_{x_j}(y), \phi_{x_i}(y))\,\mathrm{d}y = \int_{\Omega_\delta} G(x_i + y, \phi_\delta(y), \phi(y))\,\mathrm{d}y\,,$$

where $\delta = x_j - x_i$ and $\Omega_\delta = \operatorname{supp}(\phi) \cap \operatorname{supp}(\phi_\delta)$. In the second equality, passing from an integral over Ω to an integral over the subset $\Omega_\delta \subseteq \Omega$ follows immediately from the sparsity assumption (3.1b). For each fixed x_i, the affine structure of the identity above may be illuminated by collecting each contributing translation δ into the set $\mathcal{D}(x_i) = \{x_j - x_i \,:\, x_j \in \mathbb{X},\ a(\phi_{x_j}, \phi_{x_i}) \neq 0\}$ and defining a *stencil function*

$$\Phi_i^\delta(x) = \int_{\Omega_\delta} G(x+y, \phi_\delta(y), \phi(y))\,\mathrm{d}y \qquad \text{for each } \delta \in \mathcal{D}(x_i). \tag{3.3}$$

We have just reduced the computation of any $a(\phi_{x_j}, \phi_{x_i})$ to the evaluation of scalar-valued functions enumerated by affine coordinates $(x_i, x_j - x_i)$. Indeed,

$$a(\phi_{x_j}, \phi_{x_i}) = \begin{cases} \Phi_i^\delta(x_i)\,, & \text{if } \delta = x_j - x_i \in \mathcal{D}(x_i), \\ 0\,, & \text{otherwise.} \end{cases}$$

In the present scenario, there may be a different set of translations $\mathcal{D}(x_i)$ for every point x_i. However, if each point is drawn from a point lattice, most of the sets $\mathcal{D}(x_i)$ are identical. This observation is the subject of the following subsection and a foundational principle in our approach.

Remark 3.1. In the scenario that the bilinear form is symmetric, $a(u,v) = a(v,u)$, it is natural to assume that the stencil functions (3.3) will inherit a similar symmetry. Indeed, under the equivalent symmetry condition $G(x,u,v) = G(x,v,u)$ almost everywhere, one may easily verify that if $\delta = x_j - x_i$, then

$$\Phi_i^\delta(x_i) = \int_{\Omega_\delta} G(x_i + y, \phi(y), \phi_\delta(y))\,\mathrm{d}y = \int_\Omega G(x_j + y, \phi_{-\delta}(y), \phi(y))\,\mathrm{d}y = \Phi_j^{-\delta}(x_j) \tag{3.4}$$

or, equivalently, $\Phi_i^\delta(x_i) = \Phi_j^{-\delta}(x_i + \delta)$.

3.3. Local stencil functions and locally-structured meshes. An affine point lattice $\mathbb{L}$, from here on referred to only as a *lattice*, is a regularly spaced array of points in $\mathbb{R}^n$ where every point $x_i \in \mathbb{L}$ belongs to a neighborhood containing no other points in $\mathbb{L}$. In this paper, each (possibly finite) lattice is determined by a finite linearly independent set of translations in $\mathbb{R}^n$; i.e., $\mathbb{L} \subseteq \{\delta_0 + a_1\delta_1 + \cdots + a_l\delta_l \,:\, a_1, \ldots, a_l \in \mathbb{Z}\}$.

Assuming that the test function $\phi \in V$ is sufficiently localized and each point x_i is drawn from a lattice $\mathbb{L} \subseteq \Omega$, then each $\mathcal{D}(x_i)$ is a subset of a small number of admissible translations $\mathcal{D}(\mathbb{L}) = \bigcup\{\mathcal{D}(x_i) : x_i \in \mathbb{L}\}$, determined solely by the lattice structure. In such a scenario, every stencil function is closely related; i.e., $\Phi_i^\delta(x) = \Phi_j^\delta(x)$, whenever both are defined. Therefore, it is prudent to drop the subscript and define only one common stencil function $\Phi^\delta(x)$ for each $\delta \in \mathcal{D}(\mathbb{L})$. Clearly,

$$a(\phi_{x_j}, \phi_{x_i}) = \begin{cases} \Phi^\delta(x_i)\,, & \text{if } \delta = x_j - x_i \in \mathcal{D}(\mathbb{L}), \\ 0\,, & \text{otherwise.} \end{cases} \tag{3.5}$$

We are interested in exploiting (3.5) for solving a wide variety of PDEs with curvilinear geometries. Toward this end, the following examples help motivate our construction further.

REMARK 3.2. *From now on, it is useful to let V be a closed subset of $W^{1,p}(\Omega)$, for some $1 < p < \infty$. This will allow us to use a basis for $V_h \subseteq V$ consisting of finite element vertex functions [23]. Other problems where $V \subseteq \left[W^{1,p}(\Omega)\right]^k$, $k \in \mathbb{N}$, can be handled similarly (see, e.g., Subsection 4.2), by employing a basis consisting of the same vertex functions in each component.*

3.3.1. The one-dimensional setting. Let $V = H_0^1(\Omega)$, where $\Omega = (0,1) \subseteq \mathbb{R}$, and fix a small translation $dx = 1/(N+1)$. Consider the scenario where each point, $x_i = x_{i-1} + dx$, evenly divides Ω and ϕ is the piecewise-linear hat function defined $\phi(x) = \max(1 - \frac{|x|}{dx}, 0)$. Let $V_h = \{v \in H_0^1(\Omega) : v|_t \in \mathcal{P}_1(t),\ t = (x_i, x_{i+1}),\ \text{for each } 1 \le i \le N\}$ and identify each shifted hat function with the standard basis, $\phi_{x_i} = \phi_i \in V_h$. Here, we may define $\mathbb{L} = \{x_i\}$. In this case, for each $i \ge 1$, the value $a(\phi_i, \phi_j)$ can either be computed directly from (3.1a), in the standard way, or evaluated using (3.5), assuming that each Φ^δ is available at the onset of computation. Note that $\mathcal{D}(x_1) = \{0, dx\}$ and $\mathcal{D}(x_N) = \{-dx, 0\}$, but $\mathcal{D}(x_i) = \{-dx, 0, dx\}$ for each $2 \le i \le N-1$. Therefore, $\mathcal{D}(\mathbb{L}) = \{-dx, 0, dx\}$. Ultimately, defining each structured stencil function $\Phi^\delta(x) = \Phi((x_i, \delta); x)$ from an arbitrary candidate point x_i, one may verify that

$$a(\phi_j, \phi_i) = \begin{cases} \Phi^\delta(x_i)\,, & \text{if } \delta = x_j - x_i \in \{dx, 0, dx\}, \\ 0\,, & \text{otherwise,} \end{cases}$$

which is clearly the same format as (3.5).

Recall Remark 3.1. If $a(\cdot,\cdot)$ is symmetric, then, by (3.4), $\Phi^{dx}(x_i) = \Phi^{-dx}(x_i + dx)$ and the number of required stencil functions can be reduced to two. In some situations—e.g., when the zero row sum property can be employed (see Section 5)—only a single stencil function is actually required.

3.3.2. Locally-structured meshes with triangles. Let $m \in \mathbb{N}$. Beginning with a scaled Cartesian lattice $\mathbb{L}_m = 2^{-m}\mathbb{Z}^n$, it is useful to define its intersection with the closure of the right-angled reference simplex $\hat{T} = \{\hat{x} \in \mathbb{R}^n : \|\hat{x}\|_1 \le 1,\ \hat{x}_i \ge 0,\ \forall i = 1, \ldots, n\}$. This simplicial lattice, $\hat{T}_m = \mathbb{L}_m \cap \hat{T}$, can easily be transformed into a similar simplicial lattice T_m for any arbitrary simplex $T \subseteq \Omega$ via an affine transformation (see, e.g., Figure 1). Indeed, fixing the unique $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ such that $T = \{A\hat{x} + b : \hat{x} \in \hat{T}\}$, the corresponding lattice is clearly $T_m = \{A\hat{x}_i + b : \hat{x}_i \in \hat{T}_m\}$. Note that there are no interior points, $\mathring{T}_m = T_m \cap T = \emptyset$, if $m < 2$. We also define the set of boundary points $\partial T_m = T_m \setminus \mathring{T}_m$, which is always non-empty.

Let $\Omega \subseteq \mathbb{R}^n$, where $n = 2, 3$. The utility of this transformation is evident upon considering a *macro*-triangulation of Ω, say $\mathcal{T}_H$, where each *macro*-element $T \in \mathcal{T}_H$ is endowed with a simplicial lattice T_m, as defined above. Here, $H = \max_{T\in\mathcal{T}_H} H_T$, where each $H_T = \text{diam}(T)$ denotes the diameter of T. Notice that, for any fixed level $m \ge 0$, all interface points $x_i \in \partial T_m \cap \Omega$ are coincident with an interface lattice point on some neighbouring simplex. We may now define the set of all vertices on level m:

$$\mathbb{X}_m = \bigcup \{T_m : T \in \mathcal{T}_H\}.$$

For every $\mathbb{X}_m$, there is a corresponding finite element mesh such that each vertex function within a fixed macro-element $T \in \mathcal{T}_H$ is self-similar. In the cases $n = 2$ or 3, we are left to define each triangle or tetrahedron whose vertices coincide with points in $\mathbb{X}_m$. Let $[y_0, y_1, \ldots, y_k] \subseteq \mathbb{R}^n$ denote the convex combination of the points $y_0, y_1, \ldots, y_k \in \Omega$. When $n = 2$, the natural construction begins by considering the following uniform subdivision of a triangle $T = [y_0, y_1, y_2] \in \mathcal{T}_H$ into a set of four equal-volume triangles:

$$\mathcal{S}(T) = \left\{[y_0, \tfrac{y_0+y_1}{2}, \tfrac{y_0+y_2}{2}], [\tfrac{y_0+y_1}{2}, y_1, \tfrac{y_1+y_2}{2}], [\tfrac{y_0+y_2}{2}, \tfrac{y_1+y_2}{2}, y_2], [\tfrac{y_0+y_1}{2}, \tfrac{y_1+y_2}{2}, \tfrac{y_0+y_2}{2}]\right\}.$$

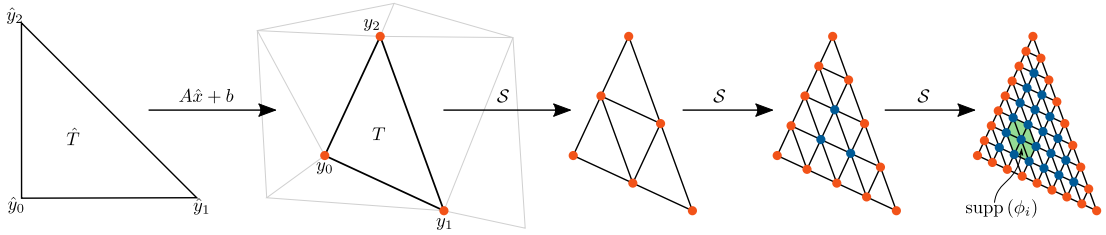

FIGURE 1. *Illustration of three refinement steps of a single macro-element T for $n = 2$ with the corresponding vertex lattices T_0, T_1, T_2, and T_3. The interior lattice points $\mathring{T}_m$ are colored blue and the boundary lattice points $\partial T_m = T_m \setminus \mathring{T}_m$ are colored in orange. Additionally, the support of an exemplary vertex function ϕ_i is shaded in green.*

For an illustration of this $n = 2$ case, see Figure 1. For $n = 3$, see the construction in [14]. Further subdivisions can then be defined recursively, *viz.*,

$$\mathcal{S}^{m+1}(T) = \bigcup \{\mathcal{S}(t) : t \in \mathcal{S}^m(T)\} \qquad \text{for all } m \geq 1\,.$$

The set of all vertices in a given subdivision $\mathcal{S}^m(T)$ forms an evenly spaced set of points inside T. The set of vertices in $\mathcal{S}^1(T)$ clearly coincides with T_1 and one can easily verify from the recursive definition that the set of vertices in $\mathcal{S}^m(T)$ also coincides with T_m. We may finally define the sequence of *locally-structured meshes*:

$$\mathcal{S}^m(\mathcal{T}_H) = \bigcup \{\mathcal{S}^m(T) : T \in \mathcal{T}_H\}, \qquad \text{for all } m \geq 1\,.$$

Notice that for each m, $\mathbb{X}_m$ coincides with the set of all vertices in $\mathcal{S}^m(\mathcal{T}_H)$. Therefore, each point x_i in $\mathbb{X}_m$ can be identified with a vertex function ϕ_i supported by only the neighboring elements [illegible] see Figure 1.

Assume that the fine mesh level, m, is chosen large enough that one may find several points in $\mathbb{X}_m$ which lie in the interior of some macro-element T. If $x_i \in \mathring{T}_m$ is any such point, then the translation set $\mathcal{D}(x_i) = \mathcal{D}(\mathring{T}_m)$ will only contain translations aligned with edges appearing in the original subdivision $\mathcal{S}^1(T)$. Identifying $\phi_{x_i} = \phi_i$ we see that for every $x_i \in \mathring{T}_m$ and $x_j \in T_m$,

$$a(\phi_j, \phi_i) = \begin{cases} \Phi_T^\delta(x_i)\,, & \text{if } \delta = (x_j - x_i) \in \mathcal{D}(\mathring{T}_m), \\ 0\,, & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\Phi_T^\delta : \mathrm{conv}(\mathring{T}_m) \to \mathbb{R}$ is a local stencil function for the current level m and macro-element T and $\mathrm{conv}(\mathring{T}_m)$ is the convex hull of $\mathring{T}_m$. In general, notice that $\Phi_T^\delta \neq \Phi_T^{2\delta}$ are two different stencil functions, corresponding to the same direction but different mesh levels.

Consider the bilinear form (3.2) with a variable diffusion coefficient. A visualization of several corresponding local stencil functions, coming from locally-structured meshes used in our numerical experiments, is given in Figure 2. It is clear from this figure that each Φ_T^δ has the potential to be a smooth function. We now come to the final essential component of our surrogate methodology; the approximation of Φ_T^δ.

REMARK 3.3. *The lattice structure of locally-structured meshes is destroyed under smooth, non-affine transformations $\hat{T} \to \widetilde{T} \subseteq \Omega$. This offers a possible impediment to our construction in the case of non-polygonal domains $\overline{\Omega} \neq \overline{\Omega}_H = \bigcup_{T \in \mathcal{T}_H} \overline{T}$. In fact, if a globally continuous transformation $\varphi : \overline{\Omega}_H \to \overline{\Omega}$ is available, with $\varphi|_T$ a smooth bijection for every $T \in \mathcal{T}_H$, then*

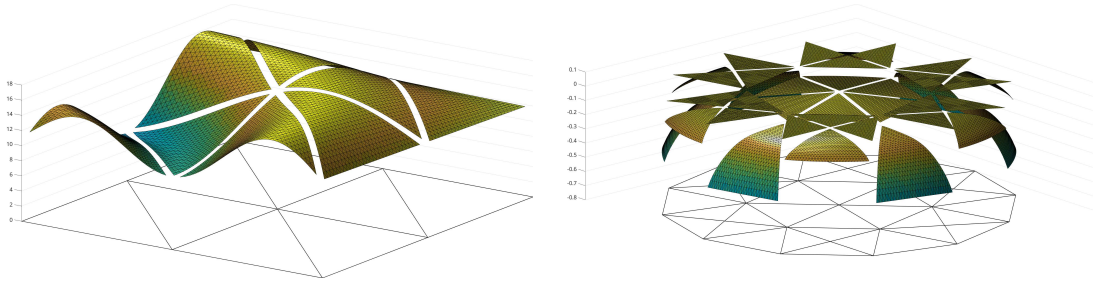

FIGURE 2. *Left: Surface plots of the local stencil functions $\Phi_T^\delta(x)$, for the degenerate direction $\delta = 0$ and level $m = 5$, from the numerical experiments recounted in Subsection 9.1.1. Here, the function is plotted over each subset $\mathrm{conv}(\mathring{T}_m) \subseteq T \in \mathcal{T}_H$. In this case, it clearly appears that each stencil function can be related to the restriction of a globally continuous function $\Phi_T^\delta(x)$. This is a result of the structure of the macro-mesh only. Right: Surface plots of the stencil functions $\Phi_T^\delta(x)$ after the first time step from the experiment recounted in Subsection 9.3, for the eastern direction δ relative to each macro-element and level $m = 5$. Moreover, although they are clearly related, it is evident that the corresponding stencil functions lack any global smoothness property.*

an equivalent method can be found using local pull-backs of φ. For example, the bilinear form in (3.2), $a_1 : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$, simply transforms to $a_{1,H} : H^1(\Omega_H) \times H^1(\Omega_H) \to \mathbb{R}$, where

$$a_{1,H}(u,v) = \int_{\Omega_H} \nabla u(x)^\top K_H(x) \nabla v(x)\, \mathrm{d}x\,, \qquad K_H = \frac{D\varphi^{-1}(K \circ \varphi) D\varphi^{-\top}}{|\det(D\varphi^{-1})|}\,. \tag{3.7}$$

Therefore, from now on, we operate under the assumption that the macro-mesh $\mathcal{T}_H$ is geometrically conforming, $\overline{\Omega} = \overline{\Omega}_H$.

3.4. The inherited regularity of stencil functions. Our approach is to locally project each stencil function Φ_T^δ in (3.6) onto a high-dimensional space of polynomials and later use this projection to compute approximate values of the stiffness matrix A. Let $\mathcal{T}_H$ be a shape-regular simplicial mesh and consider discrete functions of the variable x in the space $\mathcal{S}^m(\mathcal{T}_H)$. Before moving on, observe that definition (3.6) in fact holds for any $x_i, x_j \in T_m$ if x_i or $x_j \in \mathring{T}_m$. Therefore, due to the structure of the vertex functions in a locally-structured mesh, the domain of each Φ_T^δ can actually be extended to a set $\overline{T}_\delta$ lying between $\mathrm{conv}(\mathring{T}_m)$ and $\overline{T}$. Indeed, identifying the test function in (3.3) with the i-th vertex function, $\phi(x) = \phi_i(x + x_i)$, for an arbitrary vertex $x_i \in \mathring{T}_m$, define

$$T_\delta = \{x \in T : x + y \in T, \text{ for all } y \in \Omega_\delta = \mathrm{supp}(\phi) \cap \mathrm{supp}(\phi_\delta)\}.$$

From now on, we assume $\Phi_T^\delta : \overline{T}_\delta \to \mathbb{R}$. See Figure 3 for a depiction of the sets in a triangular mesh and note that $T_{-\delta} = T_\delta + \delta$, for every $\delta \in \mathcal{D}(T_m)$.

For any $T \in \mathcal{T}_H$, let $\mathcal{P}_q(T_\delta)$ denote the space of polynomials of degree at most q on the simplex $\overline{T}_\delta$ and let $\Pi_T^\delta : C^0(\overline{T}_\delta) \to \mathcal{P}_q(T_\delta)$ be an L^∞-continuous projection operator, $\Pi_T^\delta \circ \Pi_T^\delta = \Pi_T^\delta$. For each macro-element $T \in \mathcal{T}_H$ and level $m \in \mathbb{N}$, define the *surrogate stencil function* $\widetilde{\Phi}_T^\delta : T_\delta \to \mathbb{R}$ to be the corresponding polynomial projection of Φ_T^δ. Namely,

$$\widetilde{\Phi}_T^\delta = \Pi_T^\delta \Phi_T^\delta. \tag{3.8}$$

In order to correctly argue that a polynomial approximation of Φ_T^δ is feasible, it is necessary to classify its regularity depending on the problem at hand. In the following proposition, we show, under the modest assumptions above, that if $G(\cdot|_T, \cdot, \cdot)$ is a polynomial in its first argument, then Φ_T^δ is also a polynomial of the same degree.

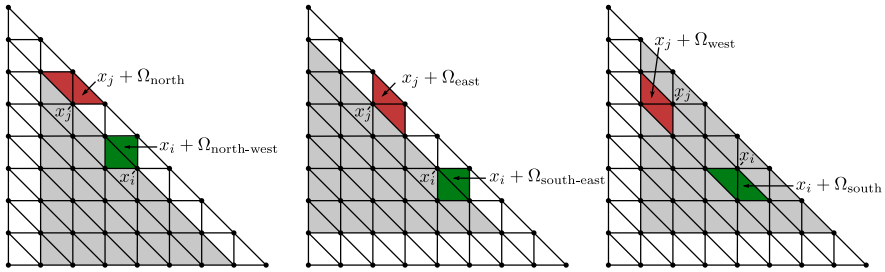

FIGURE 3. *Illustrations of the domains T_δ in gray and Ω_δ for six exemplary directions δ. Left: Northern and north-western direction. Middle: Eastern and south-eastern direction. Right: Western and southern direction.*

LEMMA 3.1. *Fix a simplex $T \in \mathcal{T}_H$. Assume that the bilinear form $a(\cdot,\cdot)$ in (2.1a) satisfies assumptions (3.1a) and (3.1b), where the integrand $G(\cdot|_T,\cdot,\cdot)$ is a polynomial of at most degree q in its first argument. Then, for any locally-structured mesh $\mathcal{S}^m(\mathcal{T}_H)$, as defined above, every local stencil function $\Phi_T^\delta \in \mathcal{P}_q(T_\delta)$ is a polynomial of the same degree.*

Proof. Recall definition (3.3). For every $x_i, x_j \in \mathbb{X}_m$, every stencil function $\Phi((x_i, x_j - x_i), x)$ is defined with a test function $\phi \in V$. Fixing any arbitrary vertex $x_i \in \mathring{T}_m$, identify this test function with the i-th vertex function, $\phi(x) = \phi_i(x + x_i)$. Let $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0$ denote a standard multi-index, $|\alpha| = \sum_{i=1}^n |\alpha_i|$ and $x^\alpha = x_1^{\alpha_1} \cdots x_t^{\alpha_n}$. By assumption, we may express $G(x, \phi_\delta(y), \phi(y)) = \sum_{|\alpha| \le l} c_\alpha(y) x^\alpha$, where each coefficient function $c(y)$ has support only in $\Omega_\delta = \operatorname{supp}(\phi) \cap \operatorname{supp}(\phi_\delta)$. Moreover, if $x + y \in T$, then

$$G(x+y, \phi_\delta(y), \phi(y)) = \sum_{|\alpha| \le l} c_\alpha(y)(x+y)^\alpha = \sum_{|\alpha| \le l} \sum_{|\nu| \le \alpha} \binom{\alpha}{\nu} c_\alpha(y) y^\nu x^{\alpha - \nu} \tag{3.9}$$

is clearly an equal degree polynomial in the variable x. The proof is completed by noting that the integral in (3.2) is performed only over the variables $y \in \Omega_\delta$, and so the stencil function acts like a convolution. Indeed, under the assumption $x \in T$, only the subset of points $x \in T_\delta = \{x \in T : x + y \in T \text{ for all } y \in \Omega_\delta\}$ guarantee that (3.9) holds at every point of integration y. In this case, by linearity of integration, Φ_T^δ is a member of $\mathcal{P}_q(T_\delta)$, with its coefficients defined by the associated integrals of the y-dependent functions in the right-hand side of (3.9). ∎

COROLLARY 3.2. *In the setting of Lemma 3.1, $\widetilde{\Phi}_T^\delta = \Phi_T^\delta$.*

Proof. Since $\Phi_T^\delta \in \mathcal{P}_q(T_\delta)$, we immediately see that $\Pi_T^\delta \Phi_T^\delta = \Phi_T^\delta$. ∎

3.5. Surrogate stiffness matrices. The main goal of the entire effort above is to guide us in reducing the vast majority of the finite element assembly process to the evaluation of a small set of functions which can, in fact, be locally approximated by polynomials. As in Subsection 3.3.2, let $\mathcal{T}_H$ be a shape-regular triangulation of a domain Ω into disjoint *macro*-simplices T.

Our construction of the surrogate matrix $\widetilde{\mathsf{A}}$ is obviously built to exploit the local lattice structure of locally-structured meshes. As argued previously, a surrogate stencil function $\widetilde{\Phi}_T^\delta$ can be used to approximate any matrix entry A_{ij} coming from a locally-structured mesh if at least one of the corresponding vertices x_i or x_j belongs to $\mathring{T}_m$. This leaves us to define only the nonzero matrix entries coming from the mutual interaction of vertex functions at the boundaries of the macro-elements. Although these entries could also be approximated by surrogate stencil functions — in this case, these additional functions would be defined on each subsimplex of the macro-mesh $\mathcal{T}_H$ — let us assume that they are computed directly. Because the growth of the macro-mesh boundary and interface interactions grow at an order of magnitude less than the

interior interactions, computing these matrix entries directly does not affect to the asymptotic performance of the methodology. Finally, letting $\partial\mathbb{X}_m = \bigcup_{T\in\mathcal{T}_H} \partial T_m$ denote the union of all macro-mesh boundary vertices, we define the general surrogate stiffness matrix

$$\widetilde{\mathsf{A}}_{ij} = \begin{cases} \int_\Omega G(y, \phi_j(y), \phi_i(y))\,\mathrm{d}y\,, & \text{if both } x_i \text{ and } x_j \in \partial\mathbb{X}_m \\ \widetilde{\Phi}^\delta_T(x_i)\,, & \text{if } \delta = (x_j - x_i) \in \mathcal{D}(\mathring{T}_m) \text{ and } x_i \text{ or } x_j \in \mathring{T}_m, \\ 0\,, & \text{otherwise.} \end{cases} \tag{3.10}$$

REMARK 3.4. *Due to the presence of the surrogate stencil functions $\widetilde{\Phi}^\delta_T$, even if A is symmetric, $\widetilde{\mathsf{A}}$ will generally not be. However, recalling (3.4), observe that if $a(\cdot,\cdot)$ is symmetric, then $\Phi^\delta_T(x) = \Phi^{-\delta}_T(x+\delta)$. Therefore, if we use related projection operators*

$$\left[\Pi^\delta_T \Phi^\delta_T\right](x) = \left[\Pi^{-\delta}_T \Phi^{-\delta}_T\right](x+\delta)\,, \tag{3.11}$$

for each opposing direction δ and $-\delta$, then $\widetilde{\mathsf{A}}$ will be symmetric. Indeed, if $\delta = x_j - x_i$, then

$$\widetilde{\mathsf{A}}_{ij} = \Pi^\delta_T \Phi^\delta_T(x_i) = \Pi^{-\delta}_T \Phi^{-\delta}_T(x_i+\delta) = \Pi^{-\delta}_T \Phi^{-\delta}_T(x_j) = \widetilde{\mathsf{A}}_{ji}\,.$$

4. Examples. In this section, we present three example problems which easily fit into the framework above.

4.1. The variable coefficient Poisson equation. Consider the Poisson-type equation $-\mathrm{div}(K\nabla u) = f$ in Ω, $u = 0$ on $\partial\Omega$, with a load $f \in L^2(\Omega)$ and a variable, symmetric positive-definite tensor K. Furthermore, assume that for each index a, b, $K_{ab} \in \mathcal{P}_q(\Omega)$. Recall (3.2) and note that we have already shown that the weak form of this problem can be cast into the framework above.

Assume that, within some set $T \subseteq \Omega$, each vertex function ϕ_i is a translation of a fixed test function $\phi(x) = \phi_i(x - x_i)$. Then for each ϕ_i, ϕ_j, the stiffness matrix entry

$$\mathsf{A}_{ij} = \int_\Omega \nabla\phi_i(x)^\top K(x) \nabla\phi_j(x)\,\mathrm{d}x \tag{4.1}$$

can equally well be expressed as the evaluation (at the point x_i) of a stencil function, which, by Lemma 3.1, is simply a polynomial of the same degree as the diffusion tensor K. In the case of locally-structured meshes, there is a locally defined stencil function $\Phi^\delta_T : \Omega \to \mathbb{R}$ for each macro-element T and level m. In this case, each $\Phi^\delta_T : \overline{T}_\delta \to \mathbb{R}$ is a polynomial (of degree at most q) on T_δ.

4.2. Linearized elasticity. Let $\vec{\nabla}$ and Div denote the row-wise distributional gradient and divergence, respectively. Now define $\epsilon(u) = \frac{1}{2}[\vec{\nabla}u + (\vec{\nabla}u)^\top]$ to be the symmetric gradient operator $\epsilon : H^1(\Omega)^n \to L^2(\Omega)^n$, where $n \geq 2$. Consider the following standard PDE model for the displacement $u \in H^1_0(\Omega)^n$ of a linearly elastic isotropic material: $-\mathrm{Div}\,\sigma = \vec{f}$, where the stress $\sigma = 2\mu\epsilon(u) + \lambda I \mathrm{div}\,u$ and the load $f \in L^2(\Omega)^n$.

The weak form of this equation is well known in the literature [20] and the associated bilinear form is simply

$$a_2(u,v) = \int_\Omega 2\mu\epsilon(u) : \epsilon(v) + \lambda \mathrm{div}(u)\,\mathrm{div}(v)\,\mathrm{d}x \qquad \text{for all } u, v \in \left[H^1_0(\Omega)\right]^n. \tag{4.2}$$

This bilinear form obviously satisfies assumptions (3.1a) and (3.1b). If we assume that the Lamé parameters $\mu, \lambda : \Omega \to \mathbb{R}$ are piecewise polynomials on a collection of disjoint subdomains $T \in \mathcal{T}_H$, then each associated stencil function is also a piecewise polynomial.

4.3. p-Laplacian diffusion. For any $1 < p < \infty$, let $\Delta_p u = \operatorname{div}(|\nabla u|^{p-2}\nabla u)$ be the p-Laplacian operator. Fix a valid parameter p and consider the nonlinear diffusion equation $\frac{\partial u}{\partial t} - \Delta_p u = f$, where $f \in L^p(\Omega)$. A simple Euler time-stepping scheme replaces the time derivative $\frac{\partial u}{\partial t}$ by the quotient $\frac{u_{k+1}-u_k}{dt}$, where $dt > 0$ is a fixed time-step parameter. Choosing backward Euler time-stepping and defining $f_k = f(k \cdot dt)$, we arrive at a semi-discrete nonlinear elliptic PDE for the solution variable u_k, which must be solved at each step $k \in \mathbb{N}$: $u_k - dt\Delta_p u_k = dt f_k + u_{k-1}$. Upon fixed point linearization of the weak form of this equation, we uncover the following bilinear form:

$$b(u,v) = \int_\Omega dt\,|\nabla \widetilde{u}|^{p-2}\,\nabla u \cdot \nabla v + u\,v\,\mathrm{d}x, \qquad \text{for all } u, v \in W^{1,p}(\Omega)\,. \tag{4.3}$$

Here, the variable coefficient $\widetilde{u} \in W^{1,p}(\Omega)$ is usually identified with the previous solution iteration in the associated fixed point algorithm (cf. Subsection 9.3).

The bilinear form $b(\cdot,\cdot)$ can easily be placed into the form of (3.1a) and each matrix entry can therefore be superceded by stencil function evaluations, $\Phi_T^\delta(x)$, as in (3.3). Alternatively, one may split $b(\cdot,\cdot)$ into a mass term $m(\cdot,\cdot)$ and a dt-weighted stiffness term $a_3(\cdot,\cdot)$. Specifically, $b(u,v) = m(u,v) + dt \cdot a_3(u,v)$, where

$$m(u,v) = \int_\Omega u\,v\,\mathrm{d}x \qquad \text{and} \qquad a_3(u,v) = \int_\Omega |\nabla \widetilde{u}|^{p-2}\,\nabla u \cdot \nabla v\,\mathrm{d}x\,. \tag{4.4}$$

With this observation in hand, we see that $b(u,v)$ may be discretized by a linear combination of independent surrogates; one for $m(\cdot,\cdot)$ and one for $a_3(\cdot,\cdot)$ (cf. Subsection 9.3). In either approach, the variable coefficient $|\nabla \widetilde{u}(x)|^{p-2}$ will generally not remain a polynomial in a subdomain of Ω and the accuracy of a surrogate stencil function $\widetilde{\Phi}_T^\delta$ will reflect the local regularity of the solution from the previous iteration, $\widetilde{u}$.

5. Boundary conditions and the zero row sum property. It is generally appropriate to define the surrogate stiffness matrix component-wise by the rule given in (3.10). Nevertheless, in some problems the operator to be discretized has a kernel which is not guaranteed to be respected by the surrogate. In such scenarios, it is possible that better performance and accuracy can be achieved if elements of this kernel are incorporated into the construction of the surrogate. This occurrence is most easily illustrated with the Poisson example from Subsection 4.1.

Consider the bilinear form $a_1 : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$, defined in (3.2). Define $V_h^{\text{ext}} = \{v \in H^1(\Omega) : v|_t \in \mathcal{P}_1(t) \text{ for each } t \in \mathcal{S}^m(\mathcal{T}_H)\}$ and $V_h = \{v \in H_0^1(\Omega) : v|_t \in \mathcal{P}_1(t) \text{ for each } t \in \mathcal{S}^m(\mathcal{T}_H)\} \subseteq V_h^{\text{ext}}$. Let the corresponding vertex function bases be $\{\phi_i\} \subseteq \{\phi_i^{\text{ext}}\}$, with $\phi_i = \phi_i^{\text{ext}}$ for $1 \le i \le N$. In most finite element software, a space like V_h^{ext} is used to impose Dirichlet boundary conditions. Indeed, a "lift" of the Dirichlet data, say $u_h^{\text{ext}} = \sum_i \mathsf{u}_i^{\text{ext}} \phi_i^{\text{ext}}$, is generally constructed from a linear combination of the set $\{\phi_i^{\text{ext}}\} \setminus \{\phi_i\}$. Then, taking $\mathsf{A}_{ij}^{\text{ext}} = a(\phi_j^{\text{ext}}, \phi_i)$ for each valid i, j, a modified load vector $\mathsf{f}^{\text{ext}} = \mathsf{f} - \mathsf{A}^{\text{ext}}\mathsf{u}^{\text{ext}}$ is used in computation.

It is obvious that $\nabla 1 = 0$ and so $a_1(1, v) = 0$ for any $v \in H^1\Omega)$. Therefore, by the partition of unity property $\sum_i \phi_i^{\text{ext}} = 1$, the zero row sum of the matrix A^{ext} also vanishes. Namely, $\sum_j \mathsf{A}_{ij}^{\text{ext}} = 0$. This property may be induced in the corresponding surrogate matrix if we simply define $\widetilde{\mathsf{A}}_{ii}^{\text{ext}} = -\sum_{j\neq i} \widetilde{\mathsf{A}}_{ij}^{\text{ext}}$, for every $x_i \in \mathbb{X}_m$, where $\widetilde{\mathsf{A}}_{ij}^{\text{ext}} = \mathsf{A}_{ij}^{\text{ext}}$ for every j where A_{ij} is not defined, and $\widetilde{\mathsf{A}}_{ij}^{\text{ext}} = \widetilde{\mathsf{A}}_{ij}$ otherwise. With this extra condition, the surrogate matrix (3.10) actually requires one fewer independent stencil function; i.e., $\Phi_T^0 = -\sum_{\delta \in \mathcal{D}(T_m)\setminus\{0\}} \Phi_T^\delta$, for every $T \in \mathcal{T}_H$. By this definition, although $\widetilde{\mathsf{A}}$ does not satisfy the zero row sum property, the matrix $\mathsf{A} - \widetilde{\mathsf{A}}$ does. Indeed,

$$\mathsf{A}_{ii} - \widetilde{\mathsf{A}}_{ii} = -\sum_{j\neq i} \left(\mathsf{A}_{ij}^{\text{ext}} - \widetilde{\mathsf{A}}_{ij}^{\text{ext}}\right) = -\sum_{j\neq i} \left(\mathsf{A}_{ij} - \widetilde{\mathsf{A}}_{ij}\right). \tag{5.1}$$

In this way, the stiffness matrix coming from linearized elasticity (4.2) is similar to the stiffness matrix coming from the Laplacian. Indeed, the zero row sum property can be incorporated into its surrogate via a straight-forward generalization.

6. Analyzing the surrogate discretization. In this section, we define and motivate what we see as some the most essential features in the analysis of our surrogate methods. We begin with a review of discrete stability in the context of (2.1c). We then touch on the concept of spectral convergence of the surrogate matrix $\widetilde{\mathsf{A}} \to \mathsf{A}$, which helps us motivate the need to control $\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max}$. This specific quantity will repeatedly appear in the *a priori* error analysis in Section 7.

6.1. Discrete stability. Let $S = \{v \in V : \|v\|_V = 1\}$ be the surface of the unit ball in V. Recall (2.2) and assume that the discretization $\mathsf{Au} = \mathsf{f}$ is stable. In the present context, this is equivalent to the existence of a constant $\alpha > 0$ such that

$$\sup_{v_h \in V_h \cap S} a(w_h, v_h) \geq \alpha \|w_h\|_V \quad \text{for all } w_h \in V_h \,.$$

Likewise, in order for the surrogate discretization $\widetilde{\mathsf{A}}\widetilde{\mathsf{u}} = \mathsf{f}$ to be stable, we must show that there exists a constant $\widetilde{\alpha} > 0$ such that

$$\sup_{v_h \in V_h \cap S} \widetilde{a}(w_h, v_h) \geq \widetilde{\alpha} \|w_h\|_V \quad \text{for all } w_h \in V_h \,. \tag{6.1}$$

Inequality (6.1) guarantees that $\widetilde{\mathsf{A}}\widetilde{\mathsf{u}} = \mathsf{f}$ has a unique solution and that $\|\widetilde{u}_h\|_V \leq \widetilde{\alpha}^{-1} \|F\|_{V^*}$. Equally important, however, it is a necessary precursor to Strang's First Lemma, which in some cases can be used, in part, to show that $\widetilde{u}_h$ converges to the exact solution u (see, e.g., Subsection 7.2).

6.2. Spectral convergence. By analyzing the singular values of A directly, (6.1) can sometimes be proven by showing that the spectrum of $\widetilde{\mathsf{A}}$ converges to the spectrum of A at a fast enough rate. The main takeaway from this section is that spectral convergence can be guaranteed by showing that $\widetilde{\mathsf{A}} \to \mathsf{A}$ in the matrix maximum norm, $\|\cdot\|_{\max}$. Before moving on, denote the k-smallest eigenvalue of a matrix $\mathsf{M} \in \mathbb{R}^{N \times N}$ as $\lambda_k(\mathsf{M})$ and let $\ell(\mathsf{M}) = \max_{1 \leq i \leq N} \#\{\mathsf{M}_{ij} \neq 0 \text{ where } 1 \leq j \leq N\}$ be the maximum number of nonzero components in M, taken across all individual rows.

PROPOSITION 6.1. *Let $\mathsf{M}, \mathsf{N} \in \mathbb{R}^{N \times N}$ be symmetric matrices. Then, for each $k = 1, \ldots, N$, it holds that*

$$|\lambda_k(\mathsf{M}) - \lambda_k(\mathsf{N})| \leq \|\mathsf{M} - \mathsf{N}\|_\infty \,.$$

The proof is placed in Appendix A. Because A and $\widetilde{\mathsf{A}}$ have the same sparsity pattern, the next result follows trivially. Note that when $n = 2$, $\ell(\mathsf{A} - \widetilde{\mathsf{A}}) \leq 7$, and when $n = 3$, $\ell(\mathsf{A} - \widetilde{\mathsf{A}}) \leq 15$, whereas $\ell(\mathsf{A})$ can be larger depending on the structure of the macro-mesh.

COROLLARY 6.2. *Let A and $\widetilde{\mathsf{A}}$ be the true and surrogate stiffness matrices in (2.2), respectively. If both A and $\widetilde{\mathsf{A}}$ are real symmetric matrices, then*

$$|\lambda_k(\mathsf{A}) - \lambda_k(\widetilde{\mathsf{A}})| \leq \ell(\mathsf{A} - \widetilde{\mathsf{A}}) \cdot \|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max} \,, \qquad k = 1, \ldots, N. \tag{6.2}$$

REMARK 6.1. *As stated above, if $\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max} \to 0$ fast enough, then Corollary 6.2 can be used in proving the stability condition (6.1). However, (6.2) is generally a very pessimistic bound and, when available, we recommend using more direct means to prove discrete stability (see, e.g., Theorem 7.1). Nevertheless, this result illustrates the importance of controlling $\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max}$, which is a central feature in all of the coming analysis.*

6.3. Controlling $\|\mathsf{A}-\widetilde{\mathsf{A}}\|_{\max}$ with the variable coefficient Poisson equation. Before we begin, some new notation is required. For any tensor $K:\Omega\to\mathbb{R}^{n\times n}$, define $\|K\|_{L^\infty(\Omega)}=\max_{a,b}\|K_{ab}\|_{L^\infty(\Omega)}$, likewise, for any $r\geq 0$, define $|K|_{W^{r+1,\infty}(T)}=\max_{a,b}|K_{ab}|_{W^{r+1,\infty}(T)}$. From now on, the notation $A\lesssim B$ will be used when two mesh-dependent quantities A and B satisfy an inequality $A\leq CB$, where C some positive H-independent constant. Likewise, when $A\lesssim B$ and $B\lesssim A$, we write $A\eqsim B$. Recall that for a macro-mesh $\mathcal{T}_H$, the diameter of a single element $T\in\mathcal{T}$ is denoted H_T and the mesh size is denoted $H=\max_{T\in\mathcal{T}_H}H_T$. We also denote the fine-scale element diameter $h_T=2^{-m}H_T$, for each $T\in\mathcal{T}_H$ and $h=2^{-m}H$.

LEMMA 6.3. *Let A and $\widetilde{\mathsf{A}}$, respectively, be the true and surrogate stiffness matrices corresponding to the bilinear form (3.2). Namely, let each component of A be given by (4.1) and each component of $\widetilde{\mathsf{A}}$ be defined by (3.10) with $G(x,u(y),v(y)):=\nabla u(y)^\top K(x)\nabla v(y)$. Fix $T\in\mathcal{T}_H$ and $0\leq r\leq q$. If $K_{ab}|_T\in W^{r+1,\infty}(T)$ for each index a,b, then*

$$\left\|\mathsf{A}-\widetilde{\mathsf{A}}\right\|_{\max,T_m}\lesssim h_T^{n-2}H_T^{r+1}|K|_{W^{r+1,\infty}(T)}\,, \tag{6.3a}$$

where $\|\mathsf{C}\|_{\max,T_m}=\max\{|\mathsf{C}_{ij}|\,:\,x_i,x_j\in T_m\}$, for any matrix C. Moreover, if each component $K_{ab}\in W^{r+1,\infty}(\mathcal{T}_H)=\prod_{T\in\mathcal{T}_H}W^{r+1,\infty}(T)$, then

$$\left\|\mathsf{A}-\widetilde{\mathsf{A}}\right\|_{\max}\lesssim h^{n-2}H^{r+1}|K|_{W^{r+1,\infty}(\mathcal{T}_H)}\,. \tag{6.3b}$$

Proof. We prove only (6.3a), (6.3b) then follows immediately. Recall (3.10) and fix $T\in\mathcal{T}_H$. Let i and j be the indices of the maximal value $|\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}|=\left\|\mathsf{A}-\widetilde{\mathsf{A}}\right\|_{\max,T_m}$. Next, because the theorem trivially holds in the degenerate case $\left\|\mathsf{A}-\widetilde{\mathsf{A}}\right\|_{\max,T_m}=0$, we proceed under the assumption that $\mathsf{A}_{ij}\neq\widetilde{\mathsf{A}}_{ij}$. Notably, it follows from Corollary 3.2 that if each $K_{ab}|_T\in\mathcal{P}_q(T)$, then $\widetilde{\mathsf{A}}_{ij}=\mathsf{A}_{ij}$ and, therefore, we find ourselves in the scenario where the diffusion tensor $K|_T$ is not a polynomial (of degree at most q). Here, we may also freely assume that $i\neq j$ because of (5.1). Indeed, for each i, $|\mathsf{A}_{ii}-\widetilde{\mathsf{A}}_{ii}|\leq\sum_{j\neq i}|\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}|\leq\ell(\mathsf{A}-\widetilde{\mathsf{A}})\cdot\max_{j\neq i}|\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}|$.

To fix notation in the remainder of the proof, we take $\phi=\phi_i$, and express

$$\widetilde{\mathsf{A}}_{ij}=\left[\Pi_T^\delta\int_{\Omega_\delta}\nabla\phi_\delta(y)^\top K(\cdot+y)\,\nabla\phi(y)\,\mathrm{d}y\right](x_i),$$

for some nonzero $\delta\in\mathcal{D}(T_m)$. Let $\mathcal{I}_T:C^0(\overline{T})\to\mathcal{P}_r(T)$ be the local Lagrange interpolant and define $\left[\mathcal{I}_T^{n\times n}K\right]_{ab}=\mathcal{I}_TK_{ab}$, for each index a,b. Splitting the two matrix entries A_{ij} and $\widetilde{\mathsf{A}}_{ij}$ into polynomial and non-polynomial parts and rewriting

$$\int_\Omega\nabla\phi_j(x)^\top\left[K-\mathcal{I}_T^{n\times n}K\right](x)\nabla\phi_i(x)\,\mathrm{d}x=\int_{\Omega_\delta}\nabla\phi_\delta(y)^\top\left[K-\mathcal{I}_T^{n\times n}K\right](x_i+y)\nabla\phi(y)\,\mathrm{d}y\,,$$
$$\int_\Omega\nabla\phi_j(x)^\top\left[\mathcal{I}_T^{n\times n}K\right](x)\nabla\phi_i(x)\,\mathrm{d}x=\int_{\Omega_\delta}\nabla\phi_\delta(y)^\top\left[\mathcal{I}_T^{n\times n}K\right](x_i+y)\nabla\phi(y)\,\mathrm{d}y\,,$$

we find that

$$\mathsf{A}_{ij}=\int_\Omega\nabla\phi_j^\top\,\mathcal{I}_T^{n\times n}K\,\nabla\phi_i\,\mathrm{d}x+\int_{\Omega_\delta}\nabla\phi_\delta(y)^\top\left[K-\mathcal{I}_T^{n\times n}K\right](x_i+y)\nabla\phi(y)\,\mathrm{d}y,$$
$$\widetilde{\mathsf{A}}_{ij}=\int_\Omega\nabla\phi_j^\top\,\mathcal{I}_T^{n\times n}K\,\nabla\phi_i\,\mathrm{d}x+\left[\Pi_T^\delta\int_{\Omega_\delta}\nabla\phi_\delta(y)^\top\left[K-\mathcal{I}_T^{n\times n}K\right](\cdot+y)\nabla\phi(y)\,\mathrm{d}y\right](x_i).$$

Upon canceling the first two terms in the expressions above, we arrive at the inequality

$$\left|\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}\right|\leq|\beta_{ij}(x_i)|+|(\Pi_T^\delta\beta_{ij})(x_i)|\,,$$

where $\beta_{ij}(x) = \int_{\Omega_\delta} \nabla\phi_\delta(y)^\top \big[K - \mathcal{I}_T^{n\times n} K\big](x+y)\nabla\phi(y)\,\mathrm{d}y$. Recall that the projection $\Pi_T^\delta : C^0(\overline{T}_\delta) \to \mathcal{P}(T_\delta)$ is continuous in the $L^\infty(\Omega)$ norm. Therefore,

$$|(\Pi_T^\delta \beta_{ij})(x_i)| \le \|\Pi_T^\delta \beta_{ij}\|_{L^\infty(T_\delta)} \lesssim \|\beta_{ij}\|_{L^\infty(T_\delta)} \lesssim \|K - \mathcal{I}_T^{n\times n} K\|_{L^\infty(T)} \|\nabla\phi_\delta \cdot \nabla\phi\|_{L^1(\Omega_\delta)}\,.$$

A standard scaling argument shows that $\|\nabla\phi_\delta \cdot \nabla\phi\|_{L^1(\Omega_\delta)} \lesssim h_T^{n-2}$. This, together with the well-known property $\|K_{ab} - \mathcal{I}_T K_{ab}\|_{L^\infty(T)} \lesssim H_T^{r+1} |K_{ab}|_{W^{r+1,\infty}(T)}$, yields the sufficient result

$$|\mathsf{A}_{ij} - \widetilde{\mathsf{A}}_{ij}| \lesssim h_T^{n-2} H_T^{r+1} |K|_{W^{r+1,\infty}(T)}\,. \qquad \square$$

REMARK 6.2. *The proof of Lemma 6.3 can be read as a blueprint which extends to the settings of the bilinear forms $a_2(\cdot,\cdot)$, $a_3(\cdot,\cdot)$, defined in (4.2) and (4.4). Indeed, when $a_2(\cdot,\cdot)$ is considered, the only significant modification to the proof above is that an interpolation operator $\mathcal{I}_T : C^0(\overline{T}) \to \mathcal{P}_r(T)$ must be introduced for each Lamé parameter μ and λ. The adaption to the setting $a(\cdot,\cdot) = a_3(\cdot,\cdot)$, is obvious. Ultimately,*

$$\big\|\mathsf{A} - \widetilde{\mathsf{A}}\big\|_{\max} \lesssim h^{n-2} H^{r+1} \cdot \begin{cases} \big|\lambda\big|_{W^{r+1,\infty}(\mathcal{T}_H)} + \big|\mu\big|_{W^{r+1,\infty}(\mathcal{T}_H)} & \text{if } a(\cdot,\cdot) = a_2(\cdot,\cdot)\,, \\ \big||\nabla\widetilde{u}|^{p-2}\big|_{W^{r+1,\infty}(\mathcal{T}_H)} & \text{if } a(\cdot,\cdot) = a_3(\cdot,\cdot)\,. \end{cases} \tag{6.4}$$

Moreover, the proof may be easily modified to permit surrogates $\widetilde{\mathsf{A}}$ without the zero row sum property. Likewise, scenarios involving fewer derivatives (which generally do not possess the zero row sum property), e.g., $a(\cdot,\cdot) = m(\cdot,\cdot)$, have similar bounds but invoke a different scaling in h.

7. *A priori* error estimation for the variable coefficient Poisson equation. In this section, we present a thorough analysis of a surrogate discretization of the variable coefficient Poisson equation. Given a load $f \in L^2(\Omega)$ and symmetric positive-definite tensor $K : \Omega \to \mathbb{R}^{n\times n}$, the corresponding weak form may be written as follows.

$$\text{Find } u \in H_0^1(\Omega) \text{ satisfying} \quad a(u,v) = F(v) \quad \text{for all } v \in H_0^1(\Omega)\,, \tag{7.1}$$

where $a(u,v) = \int_\Omega \nabla u^\top K \nabla v\,\mathrm{d}x$ and $F(v) = \int_\Omega fv\,\mathrm{d}x$. As done in Section 5, define $V_h = \{v \in H_0^1(\Omega) : v|_t \in \mathcal{P}_1(t) \text{ for each } t \in \mathcal{S}^m(\mathcal{T}_H)\}$ and let the corresponding vertex function basis be $\{\phi_i\}$.

7.1. Coercivity. In the present setting, observe that each $v_h \in V_h$ can be expressed as $v_h(x) = \sum_i v_h(x_i)\phi_i(x)$. Therefore, due to the zero row/column sum property (5.1), we find

$$\begin{aligned} \widetilde{a}(v_h, w_h) - a(v_h, w_h) &= \sum_{i,j} \big(\widetilde{\mathsf{A}}_{ij} - \mathsf{A}_{ij}\big) v_h(x_i) w_h(x_j) \\ &= \frac{1}{2} \sum_{i\neq j} \big(\mathsf{A}_{ij} - \widetilde{\mathsf{A}}_{ij}\big)(v_h(x_i) - v_h(x_j))(w_h(x_i) - w_h(x_j))\,. \end{aligned} \tag{7.2}$$

Due to the mutual sparsity of the matrices $\widetilde{\mathsf{A}}$ and A, every nonzero term in the sum above can be rewritten as $\big(\mathsf{A}_{ij} - \widetilde{\mathsf{A}}_{ij}\big)(v_h(x_i) - v_h(x_i+\delta))(w_h(x_i) - w_h(x_i+\delta))$, for some nonzero δ. Because $|\delta| \eqsim h$ by construction, one easily arrives at the following upper bound:

$$a(v_h, w_h) - \widetilde{a}(v_h, w_h) \lesssim h^{2-n} \big\|\mathsf{A} - \widetilde{\mathsf{A}}\big\|_{\max} \|\nabla v_h\|_0 \, \|\nabla w_h\|_0\,. \tag{7.3}$$

We now arrive at the main result of this subsection.

THEOREM 7.1. *Let $0 \le r \le q$. Assume that $a(\cdot,\cdot)$ is coercive and $K \in \left[W^{r+1,\infty}(\mathcal{T}_H)\right]^{n\times n}$. Then, for any fine enough macro-mesh $\mathcal{T}_H$, the surrogate bilinear form $\widetilde{a} : V_h \times V_h \to \mathbb{R}$ is also coercive.*

Proof. Let $S = \{v \in H^1 : \|v\|_1 = 1\}$ be the surface of the unit ball in H^1. Recall that since $a(\cdot,\cdot)$ is coercive, there exists a coercivity constant $\alpha > 0$ such that $a(v,v) \ge \alpha$ for all $v \in S$. Notice that $\alpha \le a(v_h,v_h) \le \widetilde{a}(v_h,v_h) + |a(v_h,v_h) - \widetilde{a}(v_h,v_h)|$ for all $v_h \in V_h \cap S$ and, therefore,

$$\alpha - |a(v_h,v_h) - \widetilde{a}(v_h,v_h)| \le \widetilde{a}(v_h,v_h) \quad \text{for all } v_h \in V_h \cap S\,.$$

Here, the second term on the left may be bounded from above using (7.3) and Lemma 6.3 as follows,

$$|a(v_h,v_h) - \widetilde{a}(v_h,v_h)| \lesssim h^{2-n}\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max}\|\nabla v_h\|_0^2 \lesssim H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}\,.$$

Thus, for any small enough H, we see that $0 < \alpha - |a(v_h,v_h) - \widetilde{a}(v_h,v_h)| \le \widetilde{a}(v_h,v_h)$, as necessary. ∎

7.2. Convergence of the surrogate solution in the H^1 norm. The purpose of this subsection is to derive a mesh-dependent upper bound on the error in the surrogate solution $\widetilde{u}_h$ of the form $\|u - \widetilde{u}_h\|_1 \le C(K,\Omega,u)h + \widetilde{C}(K,\Omega,u)H^{r+1}$. In doing so, we choose to emphasize the primary difference from the classical $\|u - u_h\|_1 \le C(K,\Omega,u)h$ error estimate by absorbing the coercivity and continuity constants (which depend on both K and Ω) into the $\lesssim$ symbol. We begin with a particular version of the First Strang Lemma [37].

LEMMA 7.2. *Let $S = \{v \in H^1 : \|v\|_1 = 1\}$ be the surface of the unit ball in H^1. Assume that $\widetilde{a} : V_h \times V_h \to \mathbb{R}$ is coercive. The following error estimate holds for the surrogate solution $\widetilde{u}_h$ of the variable coefficient model problem (7.1):*

$$\|u - \widetilde{u}_h\|_1 \lesssim \inf_{w_h \in V_h}\left[\|u - w_h\|_1 + \sup_{v_h \in V_h \cap S}|\widetilde{a}(w_h,v_h) - a(w_h,v_h)|\right]. \tag{7.4}$$

THEOREM 7.3. *Let $0 \le r \le q$ and assume that $K \in \left[W^{r+1,\infty}(\Omega)\right]^{n\times n}$ is symmetric and positive definite with $\lambda_1(K)$ bounded away from zero almost everywhere. Let $u \in H^1(\Omega)$ and $\widetilde{u}_h \in V_h$ be the unique solutions to (2.1a) and (2.1c), respectively, where $a(u,v) = \int_\Omega \nabla u^\top K \nabla v \,\mathrm{d}x$ and $F(v) = \int_\Omega f v \,\mathrm{d}x$. Then, for any sufficiently fine macro-mesh $\mathcal{T}_H$, the following upper bound holds:*

$$\|u - \widetilde{u}_h\|_1 \lesssim h|u|_2 + H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}|u|_1\,.$$

Proof. With the assumptions above, $a(\cdot,\cdot)$ is coercive. Therefore, by Theorem 7.1, if the macro-mesh $\mathcal{T}_H$ is taken fine enough, then $\widetilde{a} : V_h \times V_h \to \mathbb{R}$ is coercive. We now bound the right-hand side of (7.4). Invoking (7.3), we find

$$\|u - \widetilde{u}_h\|_1 \lesssim \|u - w_h\|_1 + h^{2-n}\left\|\mathsf{A} - \widetilde{\mathsf{A}}\right\|_{\max}\|\nabla w_h\|_0\,,$$

for every $w_h \in V_h$. Setting $w_h = \mathcal{S}\mathcal{Z}_h u$, the Scott–Zhang interpolant of u [36], we see that

$$\|u - \widetilde{u}_h\|_1 \lesssim \|u - \mathcal{S}\mathcal{Z}_h u\|_1 + h^{2-n}\left\|\mathsf{A} - \widetilde{\mathsf{A}}\right\|_{\max}|\mathcal{S}\mathcal{Z}_h u|_1 \lesssim h|u|_2 + h^{2-n}\left\|\mathsf{A} - \widetilde{\mathsf{A}}\right\|_{\max}|u|_1\,.$$

In order to finish the proof, recall that $h^{2-n}\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max} \lesssim H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}$, by Lemma 6.3. ∎

7.3. Convergence of the surrogate solution in the L^2 norm. In this subsection, we prove an L^2 error estimate of the form $\|u-\widetilde{u}_h\|_0 \le C(K,\Omega,u)h^2 + \widetilde{C}(K,\Omega,u)H^{r+1}$. A second result, which elicits accelerated H-convergence, is also proved under the additional assumption $\sum_{x_i,\in T_m^\delta}\big[\Phi_T^\delta - \Pi_T^\delta\Phi_T^\delta\big](x_i) = 0$, where $T_m^\delta = T_m \cap \overline{T}_\delta$. This is a property which naturally arises for the specific class of least-squares projections introduced in Subsection 8.1. Again, we emphasize the primary differences from the corresponding classical error estimate by absorbing the standard constants into the $\lesssim$ symbol.

THEOREM 7.4. *Under the conditions of Theorem 7.3, if $\Omega\subseteq\mathbb{R}^n$ is a convex domain, then the following additional upper bound on the error in the surrogate solution holds:*

$$\|u-\widetilde{u}_h\|_0 \lesssim h^2|u|_2 + H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}|u|_1\,. \tag{7.5a}$$

Moreover, if $r>0$ and $\sum_{x_i\in T_m^\delta}\big[\Phi_T^\delta - \Pi_T^\delta\Phi_T^\delta\big](x_i)=0$, for each $T\in\mathcal{T}_H$ and $\delta\in\mathcal{D}(\mathring{T}_m)$, then

$$\|u-\widetilde{u}_h\|_0 \lesssim h^2|u|_2 + H^{r+2}|K|_{W^{r+1,\infty}(\Omega)}\|\nabla u\|_1\,. \tag{7.5b}$$

Proof. By the triangle inequality, $\|u-\widetilde{u}_h\|_0 \le \|u-u_h\|_0 + \|u_h-\widetilde{u}_h\|_0$, where $u_h\in V_h$ is the discrete solution coming from (2.1b). It can be shown that if Ω is convex, then $u\in H^2(\Omega)\cap H_0^1(\Omega)$; see, e.g., [27]. It then follows from standard arguments that $\|u-u_h\|_0\lesssim h^2|u|_2$; see, e.g., [16, Theorem 5.7.6]. Therefore, we only need to analyze the term $\|u_h-\widetilde{u}_h\|_0$. Since, $\|u_h-\widetilde{u}_h\|_0\le\|u_h-\widetilde{u}_h\|_1$, proceeding as in the proof of Theorem 7.3, we quickly arrive at (7.5a).

In order to prove (7.5b), first define $w_h\in V_h$ satisfying $a(w_h,v_h) = (u_h-\widetilde{u}_h,v_h)_\Omega$, for all $v_h\in V_h$. Observe that the exact solution of the problem $a(w,v)=(u_h-\widetilde{u}_h,v)_\Omega$, for all $v\in H_0^1(\Omega)$, belongs to the space $H^2(\Omega)\cap H_0^1(\Omega)$, $\|w\|_2\lesssim\|u_h-\widetilde{u}_h\|_0$. Moreover,

$$\begin{aligned}\|u_h-\widetilde{u}_h\|_0^2 &= a(w_h,u_h-\widetilde{u}_h) = \widetilde{a}(w_h,\widetilde{u}_h) - a(w_h,\widetilde{u}_h)\\ &= \frac{1}{2}\sum_{i\neq j}(\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij})(\widetilde{u}_h(x_i)-\widetilde{u}_h(x_j))(w_h(x_i)-w_h(x_j))\,,\end{aligned}$$

where the final line follows from (7.2). As remarked previously, each nonzero term in this sum can be written as $\big(\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}\big)(\widetilde{u}_h(x_i)-\widetilde{u}_h(x_i+\delta))(w_h(x_i)-w_h(x_i+\delta))$, for some $T\in\mathcal{T}_H$ and nonzero $\delta\in\mathcal{D}(T_m)$. We can make better use of this expression with the identity $v_h(x_i)-v_h(x_i+\delta) = \nabla v_h(y_{i,\delta})$, wherein each $y_{i,\delta}$ is chosen from the edge connecting x_i and $x_i+\delta$, and with the relationship $\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij} = \big[\Phi_T^\delta-\Pi_T^\delta\Phi_T^\delta\big](x_i)$, since $\mathsf{A}_{ij}-\widetilde{\mathsf{A}}_{ij}\neq 0$ and $i\neq j$. With these observations in hand, we have

$$\begin{aligned}2\|u_h-\widetilde{u}_h\|_0^2 &= \sum_{T\in\mathcal{T}_H}\sum_{\delta\in\mathcal{D}(\mathring{T}_m)}\sum_{x_i\in T_m^\delta}\big[\Phi_T^\delta-\Pi_T^\delta\Phi_T^\delta\big](x_i)\,(\nabla\widetilde{u}_h(y_{i,\delta})\cdot\delta)(\nabla w_h(y_{i,\delta})\cdot\delta)\\ &\le \sum_{T\in\mathcal{T}_H}\sum_{\delta\in\mathcal{D}(\mathring{T}_m)}\sum_{x_i\in T_m^\delta}\big[\Phi_T^\delta-\Pi_T^\delta\Phi_T^\delta\big](x_i)\Big((\nabla\widetilde{u}_h(y_{i,\delta})\cdot\delta)(\nabla w_h(y_{i,\delta})\cdot\delta) - C\Big)\\ &\lesssim h^{-n}\|\mathsf{A}-\widetilde{\mathsf{A}}\|_{\max}\sum_{T\in\mathcal{T}_H}\sum_{\delta\in\mathcal{D}(\mathring{T}_m)}\|(\nabla\widetilde{u}_h\cdot\delta)(\nabla w_h\cdot\delta)-C\|_{L^1(T)}\,.\end{aligned}$$

If we set the constant above to the following average value, $C := \frac{1}{\mathrm{vol}(T)}\int_T(\nabla u\cdot\delta)(\nabla w\cdot\delta)\,\mathrm{d}x$, then $\|(\nabla u\cdot\delta)(\nabla w\cdot\delta)-C\|_{L^1(T)}\lesssim H_T|(\nabla u\cdot\delta)(\nabla w\cdot\delta)|_{W^{1,1}(T)}$. Therefore,

$$\begin{aligned}\|(\nabla\widetilde{u}_h\cdot\delta)(\nabla w_h\cdot\delta)-C\|_{L^1(T)} &\le \|(\nabla\widetilde{u}_h\cdot\delta)(\nabla w_h\cdot\delta)-(\nabla u\cdot\delta)(\nabla w_h\cdot\delta)\|_{L^1(T)}\\ &\quad+\|(\nabla u\cdot\delta)(\nabla w_h\cdot\delta)-(\nabla u\cdot\delta)(\nabla w\cdot\delta)\|_{L^1(T)}+\|(\nabla u\cdot\delta)(\nabla w\cdot\delta)-C\|_{L^1(T)}\\ &\lesssim \|\nabla(u-\widetilde{u}_h)\cdot\delta\|_{0,T}\,\|\nabla w_h\cdot\delta\|_{0,T}+\|\nabla(w-w_h)\cdot\delta\|_{0,T}\,\|\nabla u\cdot\delta\|_{0,T}\\ &\quad+H_T\|\nabla u\cdot\delta\|_{1,T}\|\nabla w\cdot\delta\|_{1,T}\,.\end{aligned}$$

After summing over all $T \in \mathcal{T}_H$ and $\delta \in \mathcal{D}(T_m)$ and taking into account $|\delta| \eqsim h$, we arrive at the bound

$$\begin{aligned}\|u_h - \widetilde{u}_h\|_0^2 &\lesssim h^{2-n}\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max}\big(\|u - \widetilde{u}_h\|_1 \, |w_h|_1 + \|w - w_h\|_1 \, |u|_1 + H\|\nabla u\|_1\|\nabla w\|_1\big)\\ &\lesssim h^{2-n}\|\mathsf{A} - \widetilde{\mathsf{A}}\|_{\max}\big(h\,|u|_2 + H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}|u|_1 + H\,\|\nabla u\|_1\big)\|w\|_2\\ &\lesssim H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}\big(H\,\|\nabla u\|_1 + H^{r+1}|K|_{W^{r+1,\infty}(\Omega)}|u|_1\big)\|w\|_2\,.\end{aligned}$$

The proof is completed by recalling that $\|w\|_2 \lesssim \|u_h - \widetilde{u}_h\|_0$. ∎

REMARK 7.1. *As stated previously, the error estimates in Theorems 7.3 and 7.4 not do track the constants appearing in the corresponding classical estimates. When $r > 0$, none of the constants in either classical estimate depend on the higher order norms $|K|_{W^{r+1,\infty}(\Omega)}$.*

REMARK 7.2. *The main ingredients in the error analysis presented in this section are Lemma 6.3, Strang's First Lemma, and (7.2), which simply follows from the zero row sum property and symmetry. Simple generalizations of Lemma 6.3, for the bilinear forms $a_2(\cdot,\cdot)$ and $a_3(\cdot,\cdot)$ have been stated in (6.4). Meanwhile, (7.2) simply follows from the zero row sum property of $\widetilde{\mathsf{A}}$ and symmetry. Therefore, we see no reason to doubt that our analysis here can be generalized to the other problems of interest in this paper. Hence, we proceed with numerical verification and demonstration.*

8. Implementation. The performance of a numerical method largely depends on its implementation. Therefore, in this section, we highlight the important features of ours. We use the HyTeG finite-element software framework [29] as the core framework for all the numerical experiments in Section 9. It offers efficient distributed data structures for simplicial meshes in 2D and 3D, which serve as a basis for the implementation of massively parallel fast iterative solvers. Its main concept is based on the idea that a coarse input mesh is split into its geometrical primitives, i.e., vertices, edges, and faces, and each of these primitives is uniformly refined. Because the primitives of the same dimension are decoupled from the others, all primitives of the same dimension may be processed in parallel. This partitioning and the hierarchy of locally structures meshes allows for efficient parallel implementations of geometric multigrid methods. More importantly, these data structures fit perfectly to the concept of macro-elements introduced in Subsection 3.3. The problems in this paper are mainly solved by employing a geometric multigrid solver using V-cycles with a hybrid Gauss–Seidel smoother. On the coarsest grid, either a preconditioned conjugate gradient method or the direct solver MUMPS [1, 2], as provided by the PETSc interface [4, 5], is used. For improved parallel scalability of the coarse grid solver, agglomeration techniques as provided by `PCTELESCOPE` [35] are used in runs with many processes.

8.1. Polynomial least squares regression. An important factor in the performance of the surrogate approach is the approximation of the stencil functions Φ_T^δ by polynomials $\widetilde{\Phi}_T^\delta$. This step in the solver process must be very fast and is usually done in a pre-process step before the actual solve. After various preparatory experiments, we have seen satisfactory performance and accuracy from simply computing $\widetilde{\Phi}_T^\delta = \Pi_T^\delta \Phi_T^\delta$ via solving a simple least-squares problem, which we now describe.

Let $T \in \mathcal{T}_H$ be a macro-element and recall that T_m is the associated lattice on level m. Suppose that Φ_T^δ is the stencil function in direction $\delta \in \mathcal{D}(T_m)$ which we want to approximate. For the least-squares regression, we fix a level m_{LS} with $m \geq m_{\mathrm{LS}} \geq 2$ and define the set of least-squares points $T_{\mathrm{LS}}^\delta \coloneqq T_{m_{\mathrm{LS}}} \cap \overline{T}_\delta$. Furthermore, let $\{p_k\}_{k=1}^M$ be a basis of $\mathcal{P}_q(T)$, the space of polynomials with maximal degree q. Assume that m_{LS} is chosen large enough such that $\big|T_{\mathrm{LS}}^\delta\big| \geq M$ and introduce the following norm on $\mathcal{P}_q(T)$: $\|p\|_{T_{\mathrm{LS}}^\delta}^2 \coloneqq \sum_{x_i \in T_{\mathrm{LS}}^\delta} p(x_i)^2$. The

least-squares regression problem, which in turn defines Π_T^δ, is formalized as follows:

$$\text{Find } \mathsf{c} \in \mathbb{R}^M \text{ satisfying} \quad \mathsf{c} = \operatorname*{arg\,min}_{\mathsf{d}\in\mathbb{R}^M} \left\| \Phi_T^\delta - \sum_{k=1}^{M} d_k p_k \right\|^2_{T_{\mathrm{LS}}^\delta} . \tag{8.1}$$

The approximated stencil function is then defined as $\widetilde{\Phi}_T^\delta := \sum_{k=1}^M c_k p_k$. This problem is equivalent to solving the possibly overdetermined linear system of equations $\mathsf{Bc} = \mathsf{f}$ in a least-squares sense, where $\mathsf{B}_{ij} = p_j(x_i)$ and $\mathsf{f}_i = \Phi_T^\delta(x_i)$ for $1 \le i \le |T_{\mathrm{LS}}^\delta|$ and $1 \le j \le N$. The choice of the polynomial basis is arbitrary. However, for an easier implementation, we employ the monomial basis, even knowing that the resulting linear system is ill conditioned. Since it is crucial to get numerically precise results, a stable solver for this problem has to be chosen. For this purpose, we apply the `colPivHouseholderQr` method from the Eigen 3.3.5 library [28], which offers a good balance between speed and accuracy. Obviously, each of these linear systems is independent of others, therefore they may be solved in parallel.

REMARK 8.1. *Taking into account* $\widetilde{\Phi}_T^\delta = \sum_{k=1}^M c_k p_k$, *the first order optimality condition for* (8.1) *can be stated as* $\sum_{x_i \in T_{\mathrm{LS}}^\delta} \left[\Phi_T^\delta - \Pi_T^\delta \Phi_T^\delta\right](x_i) = 0$. *If* $m = m_{\mathrm{LS}}$, *then the secondary assumption in Theorem* 7.4 *is satisfied and we see higher order convergence in* H, *as stated in* (7.5b). *Usually, when* m_{LS} *is close but not equal to* m, *we see preasymptotic* H*-convergence in between the two estimates given in* (7.5); *see Figure* 6.

REMARK 8.2. *In the case where the bilinear form* $a(\cdot,\cdot)$ *is symmetric, we need only approximate a single stencil function* Φ_T^δ *for both directions* δ *and* $-\delta$. *Indeed, as observed in Remark* 3.1, *the corresponding stencil functions are identical, up to a shift by* δ. *Furthermore, the symmetry requirement* (3.11), *from Remark* 3.4, *is satisfied with the projection operator defined above. Indeed, one may verify that for every* δ, $T_\delta = T_{-\delta} - \delta$. *Therefore,*

$$\left\| \Phi_T^\delta - \widetilde{\Phi}_T^\delta \right\|^2_{T_{\mathrm{LS}}^\delta} = \left\| \Phi_T^{-\delta} - \widetilde{\Phi}_T^{-\delta} \right\|^2_{T_{\mathrm{LS}}^{-\delta}} \qquad \text{and} \qquad \widetilde{\Phi}_T^\delta(x) = \widetilde{\Phi}_T^{-\delta}(x+\delta) .$$

Thus, on simplicial meshes in 2D, only four instead of seven polynomials per macro-element have to be determined and stored in memory. In some cases, where the zero row sum property holds, the number of required polynomials may be even reduced to three.

8.2. Fast polynomial evaluation. An even more important factor with respect to the performance of our implementation is the fast evaluation of the surrogate stencil functions $\widetilde{\Phi}_T^\delta$. Contrary to the computation of each $\widetilde{\Phi}_T^\delta$, which will happen only once per solve, these evaluations will be made during every matrix-vector multiplication. Therefore, the costs of evaluating the stiffness matrix entries associated to a degree of freedom, may not exceed the costs of evaluating the bilinear forms with the respective ansatz functions. In this case not only the reduction of floating point operations per degree of freedom is of importance, but also the required memory traffic has to be taken into account.

When performing a matrix-vector multiplication in HyTeG, the degrees of freedom in a macro-element are processed in a row-wise fashion as illustrated in the left of Figure 4. In each row, the stencil function may be interpreted as a 1D function. We assume without loss of generality, that the 1D stencil functions are aligned with the x-axis. This property is also inherited by the approximated stencil function. To further optimize the evaluation of the 1D polynomial, we exploit that the stencil functions have to be evaluated on a line subdivided into uniformly sized intervals of length h. Let (x_i, y_j) be a vertex node in the lattice and let $p_{y_j}(\cdot) := \widetilde{\Phi}_\delta(\cdot, y_j)$ be the approximated 1D stencil function associated to row y_j.

Assuming that we already have evaluated the stencil function p_{y_j} at a point x_i, we want to evaluate it at the next point $x_i + h$ as efficiently as possible. Since the grid points are

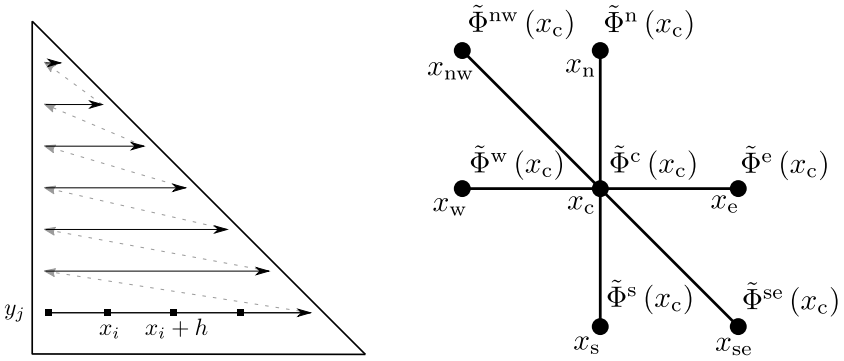

FIGURE 4. *Illustration of a loop through the degrees of freedom in a macro-element. In each row of the loop, the 2D stencil function may be interpreted as a 1D function (left). Seven stencil functions have to be evaluated in order to obtain the whole stencil for a degree of freedom (right).*

equidistantly distributed, we can use a special case of the divided differences, called forward differences [18, pg. 126].

First, we need $q+1$ helper variables $\left\{\Delta_{x_0}^{(k)}\right\}$ for $k \in \{0, 1, \ldots, q\}$ which are defined in a preprocessing step as follows:

$$\begin{aligned}\Delta_{x_0}^{(0)} &\coloneqq p_{y_j}(x_0)\,,\\ \Delta_{x_0}^{(k)} &\coloneqq \Delta_{x_0+h}^{(k-1)} - \Delta_{x_0}^{(k-1)}, \quad k \in \{1, \ldots, q\}\,.\end{aligned}$$

The value at position $p_{y_j}(x_0)$ is then given by $\Delta_{x_0}^{(0)}$. In order to obtain the value at $p_{y_j}(x_0+h)$, one has to update all the helper variables in the following way:

$$\Delta_{x_0}^{(k)} \coloneqq \Delta_{x_0}^{(k)} + \Delta_{x_0}^{(k+1)}, \quad k \in \{0, 1, \ldots, q\}\,.$$

After that, the value of $p_{y_j}(x_0+h)$ is given by $\Delta_{x_0}^{(0)}$. Doing this recursively yields the approximated stencil function values at all mesh points on a single row using only $q+1$ helper variables and $q+1$ floating point additions. When iterating through a row, in general seven polynomial evaluations, one for each direction, are required, cf. right of Figure 4. In the symmetric case this may be reduced to six polynomial evaluations, since the western stencil weight may be obtained from the previous eastern evaluation. Keep also in mind that in the symmetric case the polynomials of approximated stencil functions in opposite directions are the same but only evaluated at different positions, cf. Remark 8.2. Therefore, $6 \cdot (q+1)$ helper variables are required for a single row. For $q = 8$, these results in $54 \cdot 8$ bytes of memory which fits easily into a modern L1 CPU cache. When moving from one lattice point to another, $6 \cdot (q+1)$ vectorizable floating point additions have to be performed, to obtain the updated polynomial evaluations. Furthermore, in our implementation, the polynomial degree is realized as a C++ template parameter, therefore, all loops concerning the evaluation of a polynomial of a certain degree may be optimized at compile time. Since our focus lies in the theoretical analysis of the surrogate approach, thorough performance studies employing performance models should be considered beyond the scope of this paper. Similar performance studies have been carefully completed in [9, 10]. Thus, in the next section, we only report on relative run-times of the surrogate approach compared to the standard method, also implemented on HyTeG, using on-the-fly quadrature of the integrals stemming from the bilinear forms.

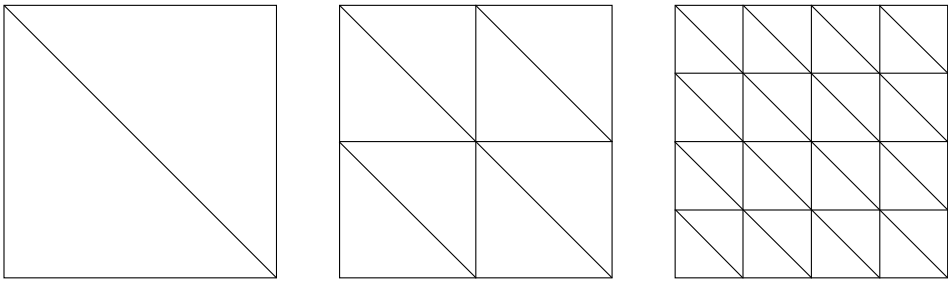

FIGURE 5. *Coarse macro-meshes of the unit-square with mesh sizes $H = H_0$ (left), $H = H_0/2$ (middle), and $H = H_0/4$ (right). Meshes with a smaller H follow the same uniform refinement pattern.*

9. Numerical experiments. In this section, we numerically verify Theorems 7.3 and 7.4, both related to the variable coefficient Poisson equation. Additionally, we present proof-of-concept results for a linearized elasticity application and a simple p-Laplacian diffusion problem. While not covered by the theory, we include these latter examples to demonstrate the breadth of generality of the methodology.

All run-time measurements in this sections were obtained on a machine equipped with two Intel® Xeon® Gold 6136 processors with a nominal base frequency of 3.0 GHz. Each processor has 12 physical cores which results in a total of 24 physical cores. The total available memory of 251 GB is split into two NUMA domains, one for each socket. We use the GCC 7.3.0 compiler and specify the following compiler arguments: `-O3 -march=native`. All the examples in this section were executed in parallel using all available 24 physical cores.

When comparing run times from the standard and the surrogate approaches, many factors are responsible for the relative speed-up of the surrogate approach. Increasing the polynomial order q of the surrogate stencil functions not only increases the run time of a multigrid iteration but also the time spent in the setup phase (i.e., computing each Φ_T^δ). The cost of the setup phase, however, is mostly dominated by the sampling level m_{LS}. Therefore, when the ratio of time spent in the iterative solver to the time spent in the setup phase(s) for solving a particular problem is large, the setup cost is almost negligible and we see the best performance. Since the problems in the following subsections differ in complexity and have different ratios of solver to setup time, the observed relative speed-ups are not directly comparable. Nonetheless, the reported speed-ups for all tested examples range between a factor of 14 and 20. Such significant speed-ups are in particular important in case of dynamic or stochastic applications. Most stochastic applications demand an enormous number of deterministic solves resulting quite often in extreme long run times. Having such a surrogate approach at hand can help to make stochastic approaches such as, e.g., multilevel Monte Carlo and its variants, more accessible for complex applications.

9.1. Quantitative benchmark problem. In this subsection, we examine the surrogate method for the variable coefficient Poisson equation which has been described and analyzed above. The strong form of the problem is

$$\begin{aligned} -\operatorname{div}(K\nabla u) &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega. \end{aligned} \tag{9.1}$$

We consider both the bilinear form coming from the scalar coefficient scenario (i.e., $K = k \cdot \mathrm{Id}$), introduced in (3.2), and the tensorial coefficient scenario, introduced in (3.7). In the scalar coefficient experiments, we use the unit-square domain $\Omega = (0,1)^2$. In the tensorial coefficient experiments, the domain Ω has a curvilinear boundary.

TABLE 1

Relative H^1 errors and experimental orders of convergence for fixed h and varying q and H in the case of problem (9.1) with the scalar coefficient (9.2). Here, the relative H^1 error with the classical FEM is $1.23 \cdot 10^{-8}$.

$\frac{H}{H_0}$	$q=1$ rel. H^1 err.	eoc	$q=2$ rel. H^1 err.	eoc	$q=3$ rel. H^1 err.	eoc	$q=4$ rel. H^1 err.	eoc
2^{-1}	$4.58 \cdot 10^{-2}$	–	$2.24 \cdot 10^{-2}$	–	$4.52 \cdot 10^{-3}$	–	$1.68 \cdot 10^{-3}$	–
2^{-2}	$1.61 \cdot 10^{-2}$	1.50	$2.75 \cdot 10^{-3}$	3.02	$4.46 \cdot 10^{-4}$	3.34	$4.70 \cdot 10^{-5}$	5.16
2^{-3}	$4.43 \cdot 10^{-3}$	1.86	$3.74 \cdot 10^{-4}$	2.88	$2.88 \cdot 10^{-5}$	3.95	$1.59 \cdot 10^{-6}$	4.89
2^{-4}	$1.15 \cdot 10^{-3}$	1.95	$4.86 \cdot 10^{-5}$	2.94	$1.84 \cdot 10^{-6}$	3.97	$5.22 \cdot 10^{-8}$	4.93
2^{-5}	$2.90 \cdot 10^{-4}$	1.98	$6.17 \cdot 10^{-6}$	2.98	$1.17 \cdot 10^{-7}$	3.98	$1.23 \cdot 10^{-8}$	2.09

TABLE 2

Relative L^2 errors and experimental orders of convergence for fixed h and varying q and H in the case of problem (9.1) with the scalar coefficient (9.2). Here, the relative L^2 error with the classical FEM is $4.10 \cdot 10^{-9}$.

$\frac{H}{H_0}$	$q=1$ rel. L^2 err.	eoc	$q=2$ rel. L^2 err.	eoc	$q=3$ rel. L^2 err.	eoc	$q=4$ rel. L^2 err.	eoc
2^{-1}	$5.75 \cdot 10^{-3}$	–	$1.92 \cdot 10^{-3}$	–	$2.90 \cdot 10^{-4}$	–	$9.46 \cdot 10^{-5}$	–
2^{-2}	$1.08 \cdot 10^{-3}$	2.42	$1.26 \cdot 10^{-4}$	3.93	$1.62 \cdot 10^{-5}$	4.16	$1.47 \cdot 10^{-6}$	6.01
2^{-3}	$1.60 \cdot 10^{-4}$	2.75	$8.84 \cdot 10^{-6}$	3.83	$5.63 \cdot 10^{-7}$	4.85	$2.47 \cdot 10^{-8}$	5.90
2^{-4}	$2.16 \cdot 10^{-5}$	2.89	$5.96 \cdot 10^{-7}$	3.89	$1.88 \cdot 10^{-8}$	4.91	$4.19 \cdot 10^{-9}$	2.56
2^{-5}	$2.80 \cdot 10^{-6}$	2.95	$3.87 \cdot 10^{-8}$	3.94	$4.10 \cdot 10^{-9}$	2.19	$4.05 \cdot 10^{-9}$	0.05

9.1.1. Scalar coefficient on unit square. In the first benchmark problem ($K = k \cdot \mathrm{Id}$), we take $\Omega = (0,1)^2$ and employ the scalar coefficient function

$$k(x,y) = \exp(xy) + \sin(3\pi xy) + \cos\left(\pi x^2 y\right) + 1 \tag{9.2}$$

in problem (9.1). The manufactured solution u is chosen as $u(x,y) = \sin(x)\sinh(y)$. The restriction of u to the boundary is chosen as Dirichlet datum g. The right-hand-side f is directly computed by inserting u into the equation. In this benchmark, we fix the finest mesh size h and report on the errors depending on H and q to show the proven $\mathcal{O}\left(H^{q+1}\right)$ estimate in the H^1 norm and $\mathcal{O}\left(H^{q+2}\right)$ in the L^2 norm. For this purpose, h is chosen to be very small in order for the error to be mostly dominated by the surrogate part.

The reference macro-mesh size is given by H_0, as illustrated in the left of Figure 5. All finer macro-meshes, with associated mesh sizes $H < H_0$, stem from uniformly refining this reference mesh; see middle and right of Figure 5. The fine mesh, with associated mesh size $h \ll H$, is the 13 times uniformly refined reference macro-mesh, i.e., $h = 2^{-13} H_0$. This fine mesh, has about $6.71 \cdot 10^7$ degrees of freedom. The approximation of the stencil functions through least-squares regression, is done on the mesh associated to mesh size $H_{\mathrm{LS}} = 2^{-8} H$. Note that this keeps the number of sampling points in each macro-element constant to 32 639. Each linear system is solved by applying geometric multigrid V(2,2) iterations until a relative residual of $1 \cdot 10^{-13}$ is obtained.

In Tables 1 and 2, the relative H^1 and L^2 errors for decreasing mesh sizes H are shown. Both tables show the expected convergence rates. In the case of the L^2 norm for $q = 3$ and $q = 4$, the convergence rate deteriorates for small macro-mesh sizes H because the discretization error is dominating the total error.

In order to show the dependence of the least-squares approach on the sampling level,

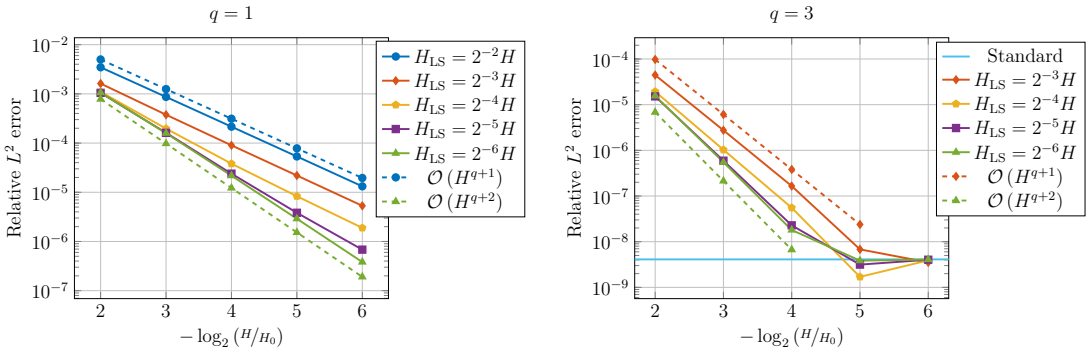

FIGURE 6. *Relative L^2 errors for fixed $h = 2^{-13}H_0$, varying H_{LS}, $q = 1$ (left), and $q = 3$ (right) in the case of the variable coefficient Poisson equation on the unit-square with a scalar coefficient. For $q = 3$ the relative L^2 error obtained from the standard approach is included, since the discretization error is dominating the surrogate error on the meshes with $H \leq 2^{-5}H_0$.*

we provide Figure 6. Here, we show two plots of the relative L^2 errors for fixed $q \in \{1, 3\}$, $h = 2^{-13}H_0$, and varying H_{LS}. From this figure, one can see that it is crucial to tune the sampling level fine enough in order to achieve optimal $\mathcal{O}\left(H^{q+2}\right)$ convergence in the L^2-norm.

REMARK 9.1. *The choice of sampling level m_{LS} or, equivalently, H_{LS} is very important, since the cost of the polynomial regression grows exponentially with m_{LS}. However, choosing a too large H_{LS} may violate the discrete L^2 projection property required in Theorem 7.4 in order to obtain an increased order of convergence. Therefore, it is crucial to choose a suitable H_{LS} for an optimal ratio between the accuracy of the solution and the run time of the stencil function approximation. In each of our experiments, setting H_{LS} two to four times larger than the fine mesh size h yielded satisfactory results with respect to accuracy and run time.*

Furthermore, in Figure 7 we want to illustrate the dependence of the polynomial degree q and the macro-mesh size H within the surrogate approach. For this purpose, we plot the central true and surrogate stencil functions over the subdomain $\{(x, y)^\top \in \Omega : x + y \geq 1\}$ for different pairings of q and H. It can be observed, that there is no visible difference of both functions when either the pairing $H = H_0$ and $q = 8$, or the pairing $H = H_0/8$ and $q = 2$ is chosen. Obviously, the quality of $\widetilde{\mathsf{A}}$ can be improved by either increasing q or decreasing H. For smooth coefficients K, increasing q is the more efficient option, like in the *hp*-FEM context.

9.1.2. Tensor coefficient on domain with curved boundaries. In the second benchmark problem, we study problem (9.1) with the symmetric and positive definite tensor coefficient

$$K(x, y) = \begin{bmatrix} 3x^2 + 2y^2 + 1 & -x^2 - y^2 \\ -x^2 - y^2 & 4x^2 + 5y^2 + 1 \end{bmatrix}. \tag{9.3}$$

Moreover, we consider the domain Ω with the curved boundary illustrated in Figure 8. In the following scenarios, $a = 0.1$ is used as the amplitude of the boundary perturbation. The mapping from the reference unit-square to the perturbed domain is defined by φ in (9.4). To map the coefficient onto the perturbed domain, we replace the coefficient K in (9.1) by a new coefficient, K_0, induced by the domain transformation, *viz.*,

$$K_0 = \frac{D\varphi^{-1}(K \circ \varphi)D\varphi^{-\top}}{|\det(D\varphi^{-1})|}, \qquad \text{where} \qquad \varphi(x, y) = \begin{bmatrix} x \\ (2ay - a)\sin^2(2\pi x) + y \end{bmatrix}. \tag{9.4}$$

The manufactured solution u is chosen to be $u(x, y) = \sin(\varphi_1(x, y))\sinh(\varphi_2(x, y))$. The restriction of u to the boundary is chosen as Dirichlet datum g and the right-hand-side f is directly

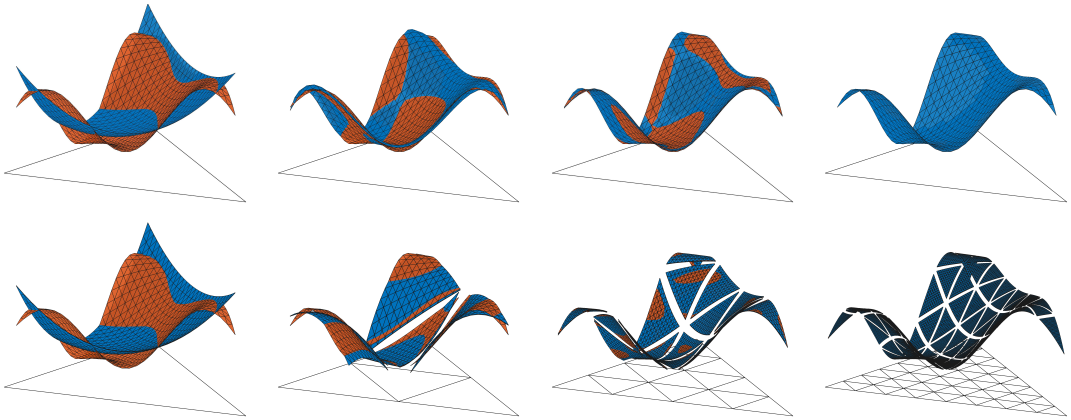

FIGURE 7. *Plots of true stencil functions in orange and surrogate stencil functions in blue for $\delta = 0$ over the subdomain $\{(x, y)^\top \in \Omega : x + y \geq 1\}$ in the case of the variable coefficient Poisson equation. Top row: Fixed $H = H_0$ and varying $q = 2$, 4, 6, and 8 from left to right. Bottom row: Fixed $q = 2$ and varying $H = H_0$, $H_0/2$, $H_0/4$, and $H_0/8$ from left to right.*

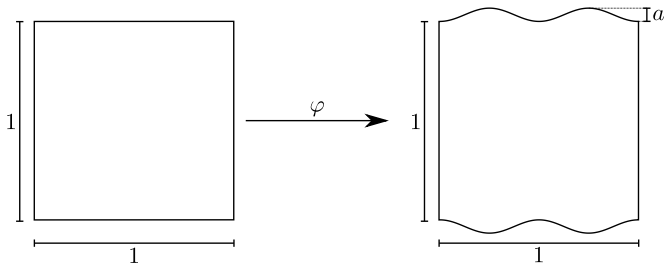

FIGURE 8. *Illustration of the mapping φ from the unit-square to the perturbed unit-square. The top boundary is parametrized by $y = a \cdot \sin(2\pi x)^2 + 1$ and the bottom boundary by $y = -a \cdot \sin(2\pi x)^2$.*

computed by inserting u into the strong form of the equation (9.1).

Here, we perform the same verification as in the previous subsection. That is, fixing h and varying q and H with the same meshes and solver settings. In Tables 3 and 4, the relative H^1 and L^2 errors for decreasing mesh sizes H are shown. Both tables show the expected convergence rates. In the case of $q = 4$, the convergence rate deteriorates for small macro-mesh sizes H, because the discretization error is dominating.

Additionally, we present results for fixed $H = 2^{-3}H_0$ and varying h and q. Table 5 shows the relative L^2 errors and convergence rates of the standard approach and the surrogate approach with $q \in \{3, 5, 7\}$. Only for $q = 7$, the L^2 error coincides with the errors from the standard approach for all h. The relative time-to-solution (rtts) shown for the surrogate approaches is defined as the time-to-solution (tts) including the setup-phase of the surrogate approach divided by the time-to-solution of the standard approach. In the case with the smallest h, the surrogate approach took at most only 7% of the time of the standard approach. That is, a speed-up by more than a factor of 14.

9.2. Linearized elasticity example. In this subsection, we compare a standard method with a surrogate method applied to the linearized elasticity problem presented in Subsection 4.2. In our surrogate method, we employ the zero row sum property described in Section 5. We choose an annular domain composed of two distinct and concentric materials under uniform

TABLE 3

Relative H^1 errors and experimental orders of convergence for fixed h and varying q and H in the case of problem (9.1) with the tensorial coefficient (9.3) and curved boundary. The relative H^1 error with the classical FEM is $1.59 \cdot 10^{-8}$.

$\frac{H}{H_0}$	$q=1$ rel. H^1 err.	eoc	$q=2$ rel. H^1 err.	eoc	$q=3$ rel. H^1 err.	eoc	$q=4$ rel. H^1 err.	eoc
2^{-1}	$1.04 \cdot 10^{-1}$	–	$7.09 \cdot 10^{-2}$	–	$2.96 \cdot 10^{-2}$	–	$9.31 \cdot 10^{-3}$	–
2^{-2}	$5.41 \cdot 10^{-2}$	0.95	$1.12 \cdot 10^{-2}$	2.67	$4.31 \cdot 10^{-3}$	2.78	$1.07 \cdot 10^{-3}$	3.12
2^{-3}	$1.37 \cdot 10^{-2}$	1.98	$2.29 \cdot 10^{-3}$	2.29	$3.14 \cdot 10^{-4}$	3.78	$4.66 \cdot 10^{-5}$	4.52
2^{-4}	$3.72 \cdot 10^{-3}$	1.89	$2.94 \cdot 10^{-4}$	2.96	$2.25 \cdot 10^{-5}$	3.80	$1.64 \cdot 10^{-6}$	4.83
2^{-5}	$9.56 \cdot 10^{-4}$	1.96	$3.77 \cdot 10^{-5}$	2.96	$1.46 \cdot 10^{-6}$	3.94	$5.55 \cdot 10^{-8}$	4.89

TABLE 4

Relative L^2 errors and experimental orders of convergence for fixed h and varying q and H in the case of problem (9.1) with the tensorial coefficient (9.3) and curved boundary. The relative L^2 error with the classical FEM is $4.56 \cdot 10^{-9}$.

$\frac{H}{H_0}$	$q=1$ rel. L^2 err.	eoc	$q=2$ rel. L^2 err.	eoc	$q=3$ rel. L^2 err.	eoc	$q=4$ rel. L^2 err.	eoc
2^{-1}	$1.44 \cdot 10^{-2}$	–	$6.43 \cdot 10^{-3}$	–	$3.08 \cdot 10^{-3}$	–	$5.41 \cdot 10^{-4}$	–
2^{-2}	$3.74 \cdot 10^{-3}$	1.95	$6.60 \cdot 10^{-4}$	3.28	$1.39 \cdot 10^{-4}$	4.47	$3.54 \cdot 10^{-5}$	3.94
2^{-3}	$5.37 \cdot 10^{-4}$	2.80	$5.75 \cdot 10^{-5}$	3.52	$6.46 \cdot 10^{-6}$	4.43	$6.13 \cdot 10^{-7}$	5.85
2^{-4}	$7.35 \cdot 10^{-5}$	2.87	$3.77 \cdot 10^{-6}$	3.93	$2.09 \cdot 10^{-7}$	4.95	$1.28 \cdot 10^{-8}$	5.58
2^{-5}	$9.52 \cdot 10^{-6}$	2.95	$2.40 \cdot 10^{-7}$	3.98	$8.05 \cdot 10^{-9}$	4.70	$4.49 \cdot 10^{-9}$	1.51

pressure loading. The problem is inspired by a similar 3D experiment documented in [25]. Let $\mathcal{B}_r \subseteq \mathbb{R}^2$ be the two-dimensional open ball of radius r with the midpoint at the origin. The computational domain is then defined as $\Omega = \mathcal{B}_{R_{\text{out}}} \setminus \mathcal{B}_{R_{\text{in}}}$. We split this domain into two disjoint sets $\Omega_I = \mathcal{B}_{R_{\text{mid}}} \setminus \mathcal{B}_{R_{\text{in}}}$ and $\Omega_O = \mathcal{B}_{R_{\text{out}}} \setminus \mathcal{B}_{R_{\text{mid}}}$ corresponding to each material. In our experiments, we fix $R_{\text{in}} = 1\,\text{cm}$, $R_{\text{mid}} = 1.75\,\text{cm}$, and $R_{\text{out}} = 2\,\text{cm}$. Refer to the leftmost diagram in Figure 9 for an illustration of the setup. Here, the macro-elements adjacent to the boundary and the material interface are mapped to the physical geometry by using the transformation described in [39].

Let $r(x) = |x|$. The strong form of the problem is

$$
\begin{aligned}
-\text{Div}\,(\sigma) &= \vec{f} && \text{in } \Omega, \\
u_\theta &= 0 && \text{on } \{(-R_{\text{out}}, 0)^\top, (0, -R_{\text{out}})^\top, (R_{\text{out}}, 0)^\top\}, \\
\sigma \cdot \hat{n} &= p_{\text{in}} \hat{e}_r && \text{on } \{x \in \partial\Omega : r = R_{\text{in}}\}, \\
\sigma \cdot \hat{n} &= -p_{\text{out}} \hat{e}_r && \text{on } \{x \in \partial\Omega : r = R_{\text{out}}\}.
\end{aligned} \tag{9.5}
$$

Here, the stress tensor σ is given by Hooke's law for isotropic materials as defined in Subsection 4.2. The unit vector in radial direction is denoted $\hat{e}_r$ and the outward pointing unit normal vector is denoted $\hat{n}$. We neglect body forces and therefore set $\vec{f} = \vec{0}$. The displacement is described in polar coordinates where u_r is the radial displacement and u_θ is the tangential displacement. In order to make the system uniquely solvable, we enforce the tangential displacement u_θ to be zero at three points; see Figure 9. The materials are chosen to be cork in the inner domain Ω_I with an A36 steel layer in the outer domain Ω_O. Poisson's ratio and Young's modulus for this scenario are $E_I = 0.02\,\text{GPa}$, $E_O = 200.0\,\text{GPa}$, $\nu_I = 0$, and $\nu_O = 0.26$. The Lamé parameters are

TABLE 5

Relative L^2 errors, experimental orders of convergence, and relative time-to-solutions for fixed H and varying q and h in the case of problem (9.1) with the tensorial coefficient (9.3) and curved boundary.

$\frac{h}{H_0}$	$\frac{H_{\mathrm{LS}}}{h}$	DoFs	standard		$q=3$			$q=5$			$q=7$		
			rel. L^2 err.	eoc	rel. L^2 err.	eoc	rtts	rel. L^2 err.	eoc	rtts	rel. L^2 err.	eoc	rtts
2^{-6}	2^{0}	$4.2 \cdot 10^{3}$	$7.14 \cdot 10^{-5}$	-	$7.15 \cdot 10^{-5}$	-	0.93	$7.14 \cdot 10^{-5}$	-	0.94	$7.14 \cdot 10^{-5}$	-	1.03
2^{-7}	2^{0}	$1.7 \cdot 10^{4}$	$1.81 \cdot 10^{-5}$	1.98	$1.87 \cdot 10^{-5}$	1.94	0.87	$1.81 \cdot 10^{-5}$	1.98	0.87	$1.81 \cdot 10^{-5}$	1.98	0.96
2^{-8}	2^{1}	$6.6 \cdot 10^{4}$	$4.55 \cdot 10^{-6}$	1.99	$6.76 \cdot 10^{-6}$	1.46	0.64	$4.55 \cdot 10^{-6}$	1.99	0.67	$4.55 \cdot 10^{-6}$	1.99	0.75
2^{-9}	2^{1}	$2.6 \cdot 10^{5}$	$1.14 \cdot 10^{-6}$	2.00	$5.70 \cdot 10^{-6}$	0.25	0.35	$1.14 \cdot 10^{-6}$	1.99	0.37	$1.14 \cdot 10^{-6}$	2.00	0.41
2^{-10}	2^{1}	$1.1 \cdot 10^{6}$	$2.85 \cdot 10^{-7}$	2.00	$6.11 \cdot 10^{-6}$	-0.10	0.15	$3.00 \cdot 10^{-7}$	1.93	0.16	$2.85 \cdot 10^{-7}$	2.00	0.19
2^{-11}	2^{1}	$4.2 \cdot 10^{6}$	$7.14 \cdot 10^{-8}$	2.00	$6.35 \cdot 10^{-6}$	-0.06	0.07	$1.16 \cdot 10^{-7}$	1.37	0.08	$7.14 \cdot 10^{-8}$	2.00	0.10
2^{-12}	2^{1}	$1.7 \cdot 10^{7}$	$1.79 \cdot 10^{-8}$	2.00	$6.47 \cdot 10^{-6}$	-0.03	0.05	$9.49 \cdot 10^{-8}$	0.29	0.06	$1.79 \cdot 10^{-8}$	2.00	0.07
2^{-13}	2^{1}	$6.7 \cdot 10^{7}$	$4.50 \cdot 10^{-9}$	1.99	$6.52 \cdot 10^{-6}$	-0.01	0.04	$9.43 \cdot 10^{-8}$	0.01	0.05	$4.54 \cdot 10^{-9}$	1.98	0.07

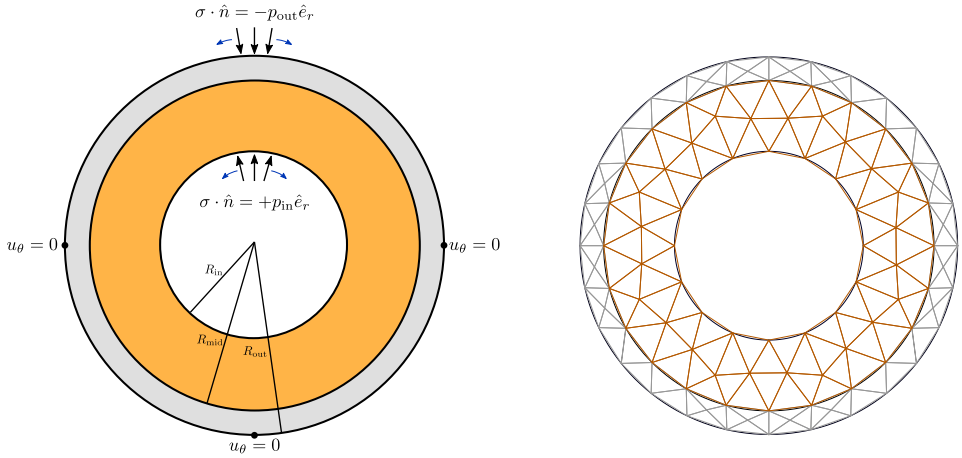

FIGURE 9. *Linear elasticity problem setup (left) and initial macro-mesh $\mathcal{T}_H$ (right).*

obtained by the expressions $\mu = \frac{E}{2(1+\nu)}$ and $\lambda = \frac{E\nu}{(1-2\nu)\cdot(1+\nu)}$. Note that while these expressions induce piecewise-constant Lamé parameters, $\lambda = \lambda(r)$ and $\mu = \mu(r)$, once the smooth domain is mapped to the computational domain depicted on the right of Figure 9, these parameters will not be piecewise-constant anymore due to the transformation. The pressure on the outer boundary is set to $p_{\mathrm{out}} = 0\,\mathrm{MPa}$ and on the inner boundary $p_{\mathrm{in}} = 1\,\mathrm{MPa}$ is prescribed.

In this particular scenario, there is an analytic solution available for the radial displacement u_r which has the form

$$u_r(r) = \begin{cases} A \cdot r + B \cdot r^{-1} & \text{if } r \in [R_{\mathrm{in}}, R_{\mathrm{mid}}]\,, \\ C \cdot r + D \cdot r^{-1} & \text{if } r \in (R_{\mathrm{mid}}, R_{\mathrm{out}}]\,. \end{cases} \tag{9.6}$$

The tangential displacement u_θ is zero everywhere due to the symmetry of the problem. In (9.6), the coefficients A, B, C, and D are uniquely determined by the following system of linear equations:

$$\begin{bmatrix} E_I R_{\mathrm{in}}^2 & E_I\,(2\nu_I - 1) & 0 & 0 \\ 0 & 0 & E_O R_{\mathrm{out}}^2 & E_O\,(2\nu_O - 1) \\ R_{\mathrm{mid}}^2 & 1 & -R_{\mathrm{mid}}^2 & -1 \\ -E_I R_{\mathrm{mid}}^2 d_O & -d_O E_I\,(2\nu_I - 1) & E_O R_{\mathrm{mid}}^2 d_I & d_I E_O\,(2\nu_O - 1) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} p_{in} R_{\mathrm{in}}^2 d_I \\ p_{out} R_{\mathrm{out}}^2 d_O \\ 0 \\ 0 \end{bmatrix},$$

where $d_I \coloneqq 2\nu_I^2 + \nu_I - 1$ and $d_O \coloneqq 2\nu_O^2 + \nu_O - 1$. This system is derived after deducing

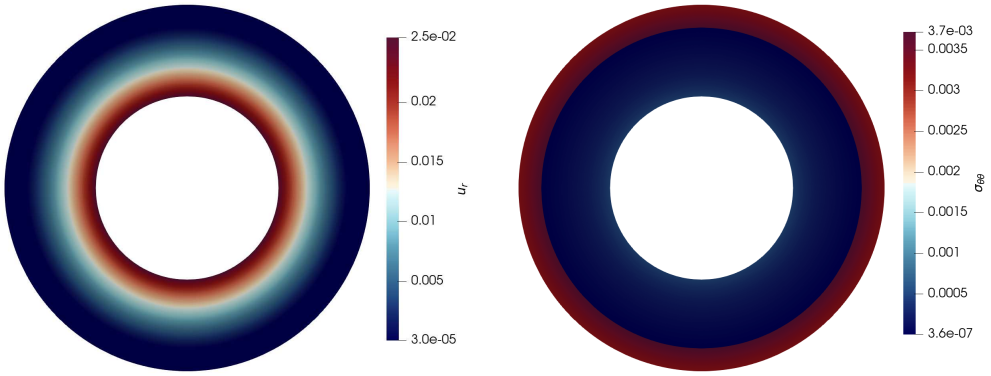

FIGURE 10. *Plots of radial displacement u_r (left) and the tangential stress $\sigma_{\theta\theta}$ (right) computed on the fine mesh $\mathcal{S}^6(\mathcal{T}_H)$, corresponding to $h = 2^{-6}H$, with $q = 4$.*

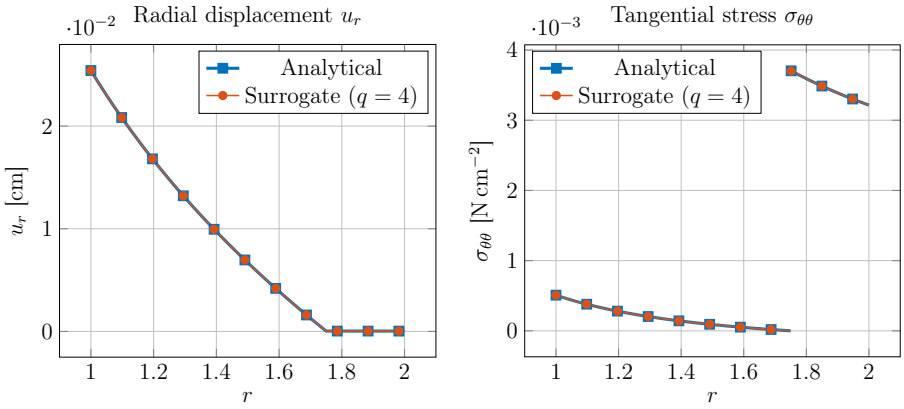

FIGURE 11. *Plots over line of the radial displacement u_r (left) and the tangential stress $\sigma_{\theta\theta}$ (right) computed on the fine mesh $\mathcal{S}^6(\mathcal{T}_H)$, corresponding to $h = 2^{-6}H$, with $q = 4$.*

the continuity of the displacement and surface traction at the material interface, then by incorporating the prescribed external forces at the boundaries.

In order to verify the accuracy of the surrogate method, we select the polynomial degree $q = 7$ and the macro-mesh $\mathcal{T}_H$, illustrated on the right of Figure 9. Note that the discontinuity in the material parameters lies along the macro-element interfaces and so $\lambda, \mu \in \prod_{T\in\mathcal{T}_H} C^\infty(T) \subsetneq W^{r+1,\infty}(\mathcal{T}_H)$, for any $r > 0$.

Each linear system is solved by applying geometric multigrid iterations with V(3,3) cycles until the relative residual is reduced by the factor $1 \cdot 10^{-7}$. On the coarsest level used in the multigrid hierarchy, we employ MUMPS as a direct solver. In Table 6, we report on the results for varying h and present the relative L^2 errors for the standard and surrogate approach, respectively. On the finest mesh involving about $1.8 \cdot 10^8$ degrees of freedom, the surrogate approach required only 5% of the time required by the standard approach while having the same accuracy. That is a speed up by a factor of 20. Figure 10 shows the radial displacement u_r and the tangential stress $\sigma_{\theta\theta}$ computed with the surrogate approach on the fine mesh $\mathcal{S}^6(\mathcal{T}_H)$, corresponding to $h = 2^{-6}H$, with $q = 4$. This is illustrated further by the plots in Figure 11 which allow a visual comparison between u_r and $\sigma_{\theta\theta}$ in the surrogate and analytical solutions.

TABLE 6

Relative L^2 errors, experimental orders of convergence, and relative time-to-solutions for fixed H, $q = 7$, and varying h in the case of the linearized elasticity problem (9.5).

			standard		$q = 7$		
$\frac{h}{H}$	$\frac{H_{\mathrm{LS}}}{h}$	DoFs	rel. L^2 err.	eoc	rel. L^2 err.	eoc	rtts
2^{-3}	2^0	$1.1 \cdot 10^4$	$3.93 \cdot 10^{-4}$	-	$3.93 \cdot 10^{-4}$	-	0.57
2^{-4}	2^0	$4.5 \cdot 10^4$	$9.66 \cdot 10^{-5}$	2.02	$9.66 \cdot 10^{-5}$	2.02	0.38
2^{-5}	2^1	$1.8 \cdot 10^5$	$2.39 \cdot 10^{-5}$	2.02	$2.39 \cdot 10^{-5}$	2.02	0.31
2^{-6}	2^2	$7.1 \cdot 10^5$	$5.92 \cdot 10^{-6}$	2.01	$5.92 \cdot 10^{-6}$	2.01	0.19
2^{-7}	2^2	$2.8 \cdot 10^6$	$1.47 \cdot 10^{-6}$	2.01	$1.47 \cdot 10^{-6}$	2.01	0.12
2^{-8}	2^2	$1.1 \cdot 10^7$	$3.68 \cdot 10^{-7}$	2.00	$3.68 \cdot 10^{-7}$	2.00	0.08
2^{-9}	2^2	$4.5 \cdot 10^7$	$9.18 \cdot 10^{-8}$	2.00	$9.19 \cdot 10^{-8}$	2.00	0.06
2^{-10}	2^2	$1.8 \cdot 10^8$	$2.30 \cdot 10^{-8}$	2.00	$2.31 \cdot 10^{-8}$	1.99	0.05

9.3. p-Laplacian diffusion example. In this subsection, we consider the time-dependent example introduced in Subsection 4.3. Here, we solve the non-linear p-Laplacian diffusion problem (9.7), given in strong form as

$$\begin{aligned} \frac{\partial u}{\partial t} - \operatorname{div}\left(|\nabla u|^{p-2} \cdot u\right) &= f \quad && \text{in } \Omega \times (0,T]\,, \\ u &= 0 \quad && \text{on } \partial\Omega \times (0,T]\,, \\ u &= u_0 \quad && \text{in } \Omega \times \{0\}\,. \end{aligned} \tag{9.7}$$

The computational domain is set to the unit disk, i.e., $\Omega \coloneqq \mathcal{B}_1$ and the right-hand-side is set to a specific constant, $f(x) = 2\hat{q}^{p/\hat{q}}$, where $\hat{q} = \frac{p}{p-1}$. The initial solution is set to $u_0(x) = 0.1 \cdot \left(1 - |x|^2\right)$. For this particular problem, the stationary limit u_∞ has an analytic solution exists with unit magnitude, namely $u_\infty(x) = 1 - |x|^{\hat{q}}$ [6, Example 3.1].

Our discretization follows a standard approach where a mass matrix $\mathsf{M}_{ij} = \int_\Omega \phi_i \phi_j \,\mathrm{d}x$ and a stiffness matrix $\mathsf{A}_{ij}(\widetilde{\mathsf{u}}) = \int_\Omega |\nabla \widetilde{u}_h|^{p-2} \nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}x$ are introduced. At this point, $\widetilde{\mathsf{u}}$ is the coefficient vector, in the $\{\phi_i\}$ basis, of an arbitrary discrete function $\widetilde{u}_h$. The time derivative is discretized by a backward Euler scheme, and the non-linearity in each time step is resolved by Picard fixed-point iterations. Let u_k^l be the coefficient vector of the discrete solution at the k-th time step and l-th fixed point iteration. Employing the bilinear form (4.3) and fixing a time step size $dt > 0$, the discrete problem in each time step $k > 0$ and fixed point iteration $l > 0$ reads as follows:

$$\left(\mathsf{M} + dt\mathsf{A}\left(\mathsf{u}_k^{l-1}\right)\right)\mathsf{u}_k^l = \mathsf{M}\mathsf{u}_{k-1} + dt\mathsf{M}\mathsf{f}\,,$$

where u_{k-1} is the final coefficient vector from the previous time step. In each time step, this system is solved multiple times (once for each fixed-point iteration) by the application of five V(2,2) multigrid cycles. The fixed-point iterations continue until the relative increment $\frac{\|\mathsf{u}_k^l - \mathsf{u}_k^{l-1}\|_2}{\|\mathsf{u}_k^l\|_2}$ is smaller than the fixed tolerance $1 \cdot 10^{-3}$. Then k is incremented and a new $\mathsf{u}_{k-1} = \mathsf{u}_{k-1}^l$ is defined.

In our surrogate method, the stencil functions of the stiffness matrix $\mathsf{A}\left(\mathsf{u}^{k-1}\right)$ are approximated by solving the least-squares problems after every fixed-point iteration, all the while enforcing the zero row sum property (cf. Subsection 9.1). Meanwhile, the stencil function of the mass matrix M is only approximated once in a pre-processing step because it does not depend on any free variables in the computation. The time step surrogate polynomials of both

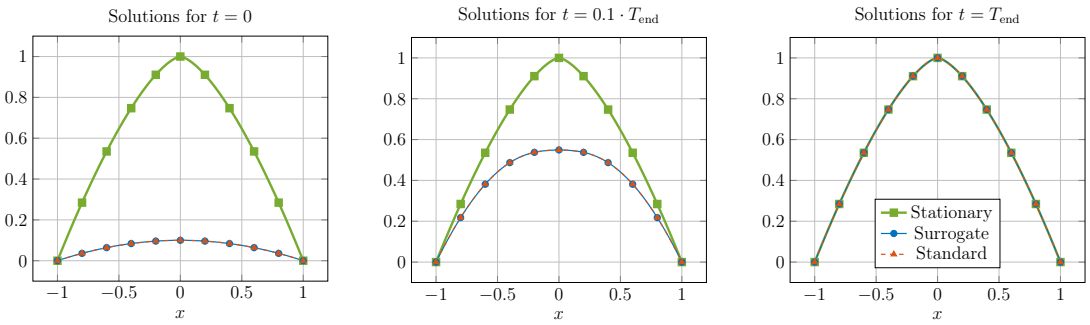

FIGURE 12. *Plots of standard, surrogate and stationary analytical solution over the line* $[0,1] \times \{0\}$ *for different times* $t = 0$, $t = 0.1 \cdot T$, *and* $t = T_{\text{end}}$.

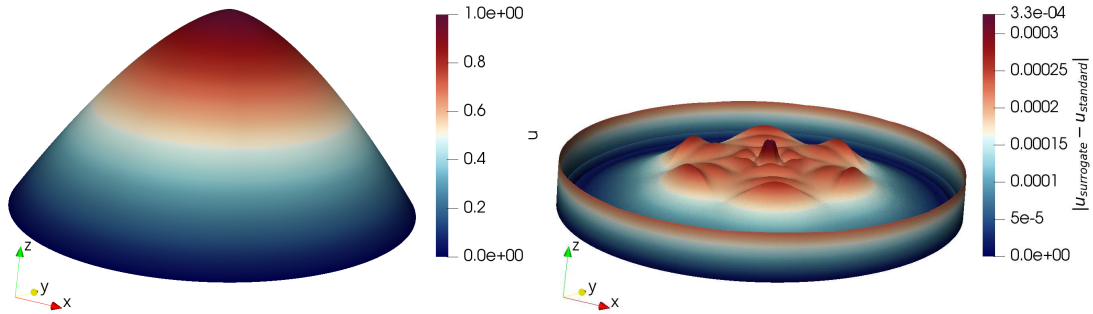

FIGURE 13. *Surface plot of non-stationary p-Laplacian surrogate solution for* $t = T_{\text{end}}$ *with* $p = 3$ *and* $q = 6$ *(left). Absolute difference between the discrete standard and surrogate solution at the time* $t = T_{\text{end}}$ *(right).*

operators are then simply summed together to obtain the time step matrix $\mathsf{M} + dt\mathsf{A}\big(\mathsf{u}^{k-1}\big)$. This particular splitting of the surrogate matrices, which reproduces the zero row sum property in the stiffness matrix, allows for faster re-approximation of the time step matrix stencil function and appears to improve the stability of the method. In this example, we did not enforce the symmetry condition featured in Subsection 3.5. Instead, whenever a vertex $x_i \in \mathbb{X}_m$ was on the boundary of a macro-element ∂T_m, we set the surrogate stiffness matrix to the exact value stiffness matrix $\widetilde{\mathsf{A}}_{ij} = \mathsf{A}_{ij}$. This minor asymmetry, is more amenable to computation because there is less data transfer and it did not affect the behavior of our multigrid solver. In fact, this choice improved our results with this problem, which we believe is due to better accuracy in the surrogate near the singularity in the coefficient; i.e., at the origin $x = (0,0)$. The success of this approach suggests that the definition given in (3.10) may be relaxed in other applications as well.[1] In the proximity of this singularity and for $p > 2$, the coefficient depending on the solution of the previous fixed-point iteration is getting very close to zero which serves as a challenge for the approximated off-diagonal stencil functions. Depending on the polynomial degree q, they might erroneously take on positive values due to overshoots which possibly results in a loss of positive definiteness of the surrogate matrix. However, this drawback could not be observed in the scenario considered in the following example.

The unit-disk is discretized by the macro-mesh $\mathcal{T}_H$ featured on the right of Figure 2. Note that the vertices of the central macro-elements meet at the origin $x = (0,0)$; i.e., exactly where the singularity occurs in the stationary limit u_∞. The simulations are conducted on

[1] See [10] for further evidence.

the mesh $\mathcal{S}^9(\mathcal{T}_H)$, which involves about $4.72 \cdot 10^6$ degrees of freedom. The macro-elements adjacent to the boundary are mapped to the physical geometry by using the mapping described in [39]. Furthermore, the least-squares regressions are carried out on the mesh corresponding to $H_{\mathrm{LS}} = 4h$ and the polynomial degree of the approximated stencil functions is fixed to $q = 6$. In this scenario, we consider the p-Laplacian operator with $p = 3$, fix the time step size $\Delta t = 1 \cdot 10^{-2}$, and solve until time $T_{\mathrm{end}} = 1$. Figure 12 illustrates the standard and surrogate solutions plotted over the line $[0, 1] \times \{0\}$ for different times t. In the left of Figure 13, the surface plot of the surrogate solution is depicted. Since the difference of the solutions is very small, we added in the right of Figure 13 a surface plot of the absolute difference of the surrogate and standard solution at the final time $t = T_{\mathrm{end}}$. The surrogate approach required only 5.4% of the time required by the standard approach. That is, a speed-up by more than a factor of 18.

Appendix A. Proofs.

Proof of Proposition 6.1. By the min-max theorem [21], the k-th eigenvalue of M is

$$\begin{aligned}\lambda_k(\mathsf{M}) &= \min_{W \subseteq \mathbb{R}^N} \left\{ \max_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{M}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = k \right\} \\ &= \max_{W \subseteq \mathbb{R}^N} \left\{ \min_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{M}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = N - k + 1 \right\}.\end{aligned}$$

Define $\mathsf{D} = \mathsf{M} - \mathsf{N}$. We first show that $\lambda_1(\mathsf{D}) \leq \lambda_k(\mathsf{M}) - \lambda_k(\mathsf{N}) \leq \lambda_N(\mathsf{D})$. Indeed,

$$\begin{aligned}\lambda_k(\mathsf{M}) &\leq \min_{W \subseteq \mathbb{R}^N} \left\{ \max_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{N}\mathsf{x} : \mathsf{x} \in W \right\} + \max_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{D}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = k \right\} \\ &\leq \min_{W \subseteq \mathbb{R}^N} \left\{ \max_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{N}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = k \right\} + \max_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{D}\mathsf{x} : \mathsf{x} \in \mathbb{R}^N \right\} \\ &= \lambda_k(\mathsf{N}) + \lambda_N(\mathsf{D})\end{aligned}$$

and, likewise,

$$\begin{aligned}\lambda_k(\mathsf{M}) &\geq \max_{W \subseteq \mathbb{R}^N} \left\{ \min_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{N}\mathsf{x} : \mathsf{x} \in W \right\} + \min_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{D}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = N - k + 1 \right\} \\ &\geq \max_{W \subseteq \mathbb{R}^N} \left\{ \min_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{N}\mathsf{x} : \mathsf{x} \in W \right\} : \dim W = N - k + 1 \right\} + \min_{\|\mathsf{x}\|_2=1} \left\{ \mathsf{x}^\top \mathsf{D}\mathsf{x} : \mathsf{x} \in \mathbb{R}^N \right\} \\ &= \lambda_k(\mathsf{N}) + \lambda_1(\mathsf{D}).\end{aligned}$$

This immediately leads us to the inequality $|\lambda_k(\mathsf{M}) - \lambda_k(\mathsf{N})| \leq \max\{|\lambda_1(\mathsf{D})|, |\lambda_N(\mathsf{D})|\}$. Now, for at least one i, $|\lambda_N(\mathsf{D})| - |\mathsf{D}_{ii}| \leq |\lambda_N(\mathsf{D}) - \mathsf{D}_{ii}| \leq \sum_{j \neq i} |\mathsf{D}_{ij}|$, by the Gershgorin circle theorem. Therefore, $|\lambda_N(\mathsf{D})| \leq \sum_j |\mathsf{D}_{ij}| \leq \|\mathsf{D}\|_\infty$. Similarly, $|\lambda_1(\mathsf{D})| \leq \|\mathsf{D}\|_\infty$. ∎

Acknowledgments. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800898. This work was also partly supported by the German Research Foundation through the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA) and by grant WO671/11-1. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (GCS, www.gauss-centre.eu) for funding this project by providing computing time on the GCS supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ, www.lrz.de).

REFERENCES

[1] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41.

[2] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet, *Hybrid scheduling for the parallel solution of linear systems*, Parallel Computing, 32 (2006), pp. 136–156.

[3] P. Arbenz, G. H. van Lenthe, U. Mennel, R. Müller, and M. Sala, *A scalable multi-level preconditioner for matrix-free μ-finite element analysis of human bone structures*, International Journal for Numerical Methods in Engineering, 73 (2008), pp. 927–947.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 3.9, Argonne National Laboratory, 2018.

[5] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.

[6] J. W. Barrett and W. B. Liu, *Finite element approximation of the p-Laplacian*, Mathematics of computation, 61 (1993), pp. 523–537.

[7] S. Bauer, D. Drzisga, M. Mohr, U. Ruede, C. Waluga, and B. Wohlmuth, *A stencil scaling approach for accelerating matrix-free finite element implementations*, arXiv preprint arXiv:1709.06793, (2017).

[8] S. Bauer, M. Huber, S. Ghelichkhan, M. Mohr, U. Rüde, and B. Wohlmuth, *Large-scale simulation of mantle convection based on a new matrix-free approach*, Preprint, (2018).

[9] S. Bauer, M. Huber, M. Mohr, U. Rüde, and B. Wohlmuth, *A new matrix-free approach for large-scale geodynamic simulations and its performance*, in International Conference on Computational Science, Springer, 2018, pp. 17–30.

[10] S. Bauer, M. Mohr, U. Rüde, J. Weismüller, M. Wittmann, and B. Wohlmuth, *A two-scale approach for efficient on-the-fly operator assembly in massively parallel high performance multigrid codes*, Applied Numerical Mathematics, 122 (2017), pp. 14–38.

[11] B. Bergen, *Hierarchical Hybrid Grids: Data Structures and Core Algorithms for Efficient Finite Element Simulations on Supercomputers: Hierarchische Hybride Gitter: Datenstrukturen und Algorithmen Zur Effizienten Simulation Mit Finiten Elementen Auf Höchstleistungsrechnern*, SCS Publishing House, 2005.

[12] B. Bergen and F. Hülsemann, *Hierarchical hybrid grids: Data structures and core algorithms for multigrid*, Numerical linear algebra with applications, 11 (2004), pp. 279–291.

[13] B. Bergen, G. Wellein, F. Hülsemann, and U. Rüde, *Hierarchical hybrid grids: Achieving TER-AFLOP performance on large scale finite element simulations*, International Journal of Parallel, Emergent and Distributed Systems, 22 (2007), pp. 311–329.

[14] J. Bey, *Tetrahedral grid refinement*, Computing, 55 (1995), pp. 355–378.

[15] J. Bielak, O. Ghattas, and E.-J. Kim, *Parallel Octree-Based Finite Element Method for Large-Scale Earthquake Ground Motion Simulation*, Computer Modeling in Engineering & Sciences, 10 (2005), pp. 99–112.

[16] S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15, Springer Science & Business Media, 2007.

[17] J. Brown, *Efficient Nonlinear Solvers for Nodal High-Order Finite Elements in 3D*, J. Scientific Computing, 45 (2010), pp. 48–63.

[18] R. L. Burden and J. D. Faires, *Numerical Analysis*, Cengage Learning, Inc, 2015.

[19] G. F. Carey and B.-N. Jiang, *Element-by-element linear and nonlinear solution schemes*, Communications in Applied Numerical Methods, 2 (1986), pp. 145–153.

[20] P. G. Ciarlet, *Three-dimensional elasticity*, vol. 20, Elsevier, 1988.

[21] R. Courant and D. Hilbert, *Methods of mathematical physics, Volume 1*, 1953.

[22] C. Engwer, R. D. Falgout, and U. M. Yang, *Stencil computations for PDE-based applications with examples from DUNE and Hypre*, Concurrency and Computation: Practice and Experience, (2017), p. 13.

[23] A. Ern and J.-L. Guermond, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media, 2013.

[24] C. Flaig and P. Arbenz, *A Highly Scalable Matrix-Free Multigrid Solver for μFE Analysis Based on a Pointer-Less Octree*, in Large-Scale Scientific Computing: 8th International Conference, LSSC 2011, Sozopol, Bulgaria, June 6-10, 2011, Revised Selected Papers, I. Lirkov, S. Margenov, and J. Waśniewski, eds., Springer Berlin Heidelberg, 2012, pp. 498–506.

[25] F. Fuentes, B. Keith, L. Demkowicz, and P. Le Tallec, *Coupled variational formulations of linear elasticity and the DPG methodology*, Journal of Computational Physics, 348 (2017), pp. 715–731.

[26] B. Gmeiner, U. Rüde, H. Stengel, C. Waluga, and B. Wohlmuth, *Towards textbook efficiency for parallel multigrid*, Numer. Math. Theor. Meth. Appl., 8 (2015), pp. 22–46.
[27] P. Grisvard, *Elliptic problems in nonsmooth domains*, Pitman Advanced Publishing Program, 1985.
[28] G. Guennebaud, B. Jacob, et al., *Eigen v3*. http://eigen.tuxfamily.org, 2010.
[29] N. Kohl, D. Thönnes, D. Drzisga, D. Bartuschat, and U. Rüde, *The HyTeG finite-element software framework for scalable multigrid solvers*, International Journal of Parallel, Emergent and Distributed Systems, (2018), pp. 1–20.
[30] M. Kronbichler and K. Kormann, *A generic interface for parallel cell-based finite element operator application*, Computers and Fluids, 63 (2012), pp. 135–147.
[31] K. Ljungkvist, *Matrix-free Finite-element Computations on Graphics Processors with Adaptively Refined Unstructured Meshes*, in Proceedings of the 25th High Performance Computing Symposium, HPC '17, Society for Computer Simulation International, 2017, pp. 1:1–1:12.
[32] K. Ljungkvist and M. Kronbichler, *Multigrid for Matrix-Free Finite Element Computations on Graphics Processors*, Tech. Rep. 2017-006, Department of Information Technology, Uppsala University, 2017.
[33] J. Loffeld and J. Hittinger, *On the arithmetic intensity of high-order finite-volume discretizations for hyperbolic systems of conservation laws*, The International Journal of High Performance Computing Applications, (2017).
[34] D. A. May, J. Brown, and L. L. Pourhiet, *A scalable, matrix-free multigrid preconditioner for finite element discretizations of heterogeneous Stokes flow*, Computer Methods in Applied Mechanics and Engineering, 290 (2015), pp. 496–523.
[35] D. A. May, P. Sanan, K. Rupp, M. G. Knepley, and B. F. Smith, *Extreme-scale multigrid components within PETSc*, in Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '16, New York, NY, USA, 2016, ACM, pp. 5:1–5:12.
[36] L. R. Scott and S. Zhang, *Finite element interpolation of nonsmooth functions satisfying boundary conditions*, Mathematics of Computation, 54 (1990), pp. 483–493.
[37] G. Strang, *Variational crimes in the finite element method*, in The mathematical foundations of the finite element method with applications to partial differential equations, Elsevier, 1972, pp. 689–710.
[38] B. van Rietbergen, H. Weinans, R. Huiskes, and B. Polman, *Computational strategies for iterative solutions of large FEM applications employing voxel data*, International Journal for Numerical Methods in Engineering, 39 (1996), pp. 2743–2767.
[39] M. Zlámal, *Curved elements in the finite element method. I*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 229–240.